PROPOSTA DE UM GERADOR DE JOGO SÉRIO PARA ENSINO DE DISCIPLINAS RELACIONADAS À ENGENHARIA DE SOFTWARE

Cleber Nardelli[1]
Centro de Educação Superior do Alto Vale do Itajaí - CEAVI
clebernardelli@gmail.com

Resumo
O processo de ensino normalmente utilizado nas cadeiras de Engenharia de Software (ES) nos cursos de graduação, expõe um quadro onde os alunos normalmente vivenciam de fato a maioria dos conceitos e práticas passadas em sala de aula, apenas no seu dia a dia de trabalho. Este cenário não ocorre somente na área de ES, tampouco na área da computação. Em parte esse fenômeno ocorre por que a maioria das instituições de ensino superior adota um currículo cuja disciplina de ES é ministrada com créditos insuficientes para cobrir todos os conceitos da área. Uma das causas pode estar relacionada a não existência de práticas educacionais alternativas as tradicionais. Na literatura encontra-se o uso de jogos como uma dessas alternativas. Quando utilizados para fins educacionais, jogos são conhecidos pelo termo "Jogo Sério". O presente trabalho tem como foco principal propor um software gerador de jogo sério, onde o professor possa mapear jogos conforme a demanda da disciplina lecionada e fazer o gerenciamento dos participantes dos jogos disponibilizados, concentrando-se neste caso em especial na disciplina de Engenharia de Software através do mapeamento completo do jogo "A Ilha dos Requisitos", voltado para a Engenharia de Requisitos.

Palavras-chave: Jogo Sério. Gerador de Jogos. Engenharia de Software.

Abstract
The teaching process typically used in the chairs of Software Engineering (ES) in undergraduate courses, presents a framework where students typically experience in fact most of the concepts and past practices in the classroom, but in their day to day work. This scenario does not occur only in the area of ES, either in computing. In part this phenomenon occurs for the majority of higher education institutions adopted a curriculum whose discipline ES is provided with insufficient credits to cover all concepts from the field. One cause may be related to lack of alternatives to traditional education practices. Is the use of games such as one of those alternatives in the literature. When used for educational purposes, games are known by the term "Serious Game". This paper focuses mainly propose a serious game generator software, where the teacher can map games on demand of taught discipline and make the management of participants of the games available, focusing this case especially in the discipline of Software Engineering through the complete mapping of the game "The Island of Requirements", facing the Requirements Engineering.

Keywords: *Serious game. Generator Games. Software Engineering.*

[1] Aluno do curso de Pós-Graduação em Engenharia de Software (UDESC).

1. Introdução

De fato os conceitos norteadores presentes na área de engenharia de software, são normalmente vistos como tediosos pelos alunos em sala de aula. "Os profissionais da área aprendem mais sobre tópicos de ES depois da graduação" (WANGENHEIM e SILVA, 2009). Os autores mencionam que "o papel de um profissional de software deixou de limitar-se somente a conhecimentos técnicos, passando a exigir um vasto conjunto de conhecimentos e habilidades como: comunicação, trabalho em equipe, etc.".

Com base no exposto, pode-se observar que o ensino das disciplinas vinculadas à engenharia de software, explora amplamente o lado conceitual deixando em segundo plano outras formas de abordagem dos conteúdos a serem repassados, como o uso de jogos sérios ou atividades lúdicas em sala de aula podendo ser citado o exemplo do desenvolvimento de aviõezinhos de papel para práticas de gestão de projetos, demonstração dos diversos ciclos de vida de desenvolvimento de software e até mesmo prototipação. Um gerador de jogos sérios poderá abranger um grande número de áreas de conhecimento dentro da engenharia de software e fornecer ao professor uma ferramenta de uso conforme demanda.

Tendo em vista o contexto supracitado pela literatura abordada em Wangenheim e Silva (2009), destaca-se a necessidade de uma forma alternativa ao método tradicional de ensino através de aulas expositivas, que explora principalmente o lúdico utilizando jogos. Segundo Tori (2010), "nos dias atuais, o sonho de todo educador é ter de seus alunos, em aula, uma fração da atenção, motivação e produtividade que esses mesmos jovens apresentam quando engajados no ato de jogar seus games preferidos.".

A utilização de jogo sério torna as atividades mais lúdicas e, consequentemente, o aprendizado pode ser mais bem assimilado pelo estudante. "A utilidade dos jogos no ensino está altamente acoplada à sua capacidade de entreter os estudantes" (THIRY et. al., 2010).

Pretende-se com o presente trabalho propor um software gerador de jogo sério, que demonstra o seu potencial através do mapeamento completo de um jogo já existente e utilizado para a prática de Engenharia de Requisitos dentro da disciplina de Engenharia de Software.

O mesmo está dividido em 8 seções. A seção 2 presenta um estudo realizado no âmbito de jogo sério em relação às práticas genéricas de atividades de engenharia de software; a seção 3 descreve uma revisão teórica e a análise de alguns trabalhos correlatos; a seção 4 descreve um estudo de caso onde um jogo já existente é mapeado pelo gerador; a seção 5 mostra as fases da construção de jogos no gerador; a seção 6 descreve o projeto do gerador de jogos; a seção 7 descreve um exemplo de um jogo construído pelo gerador de jogos e na última seção são apresentadas as considerações finais. Além disso, o trabalho conta ainda com o Apêndice A, onde são exibidos os detalhes sobre as fases para a construção de jogos no gerador; Apêndice B onde é apresentado um template próprio do documento GDD (Game Design Document, que trata-se de um plano detalhado de desenvolvimento de jogos no gerador) e Apêndice C, onde estão os documentos do caso de uso realizado e o diagrama de classes a nível de negócio.

2. Jogo Sério

Para a execução deste trabalho foi realizado inicialmente um estudo do estado da arte no âmbito de jogo sério, abordando a temática voltada principalmente para práticas genéricas de atividades de engenharia de software.

Segundo Adams (2010) apud Silva (2011), "Jogos, em geral, podem ser definidos como uma atividade lúdica, realizada no contexto de uma realidade simulada na qual os participantes tentam alcançar, pelo menos, uma meta arbitrária, não trivial, agindo de acordo com as regras". Eles utilizam o visual artístico já encontrado em outros jogos, para passar alguma mensagem de aprendizado, sendo esta cuidadosamente elaborada para ser aplicada neste sentido.

Em outra definição sobre o mesmo tema, Feijó et. al. (2010), classifica o Jogo Sério como “jogos que procuram simular desafios e situações reais referentes a um contexto, tendo muitas vezes como objetivo o treinamento ou a educação dentro de uma realidade.”.

Sobre o mesmo tema Zyda (2005) inclui na definição de jogo sério áreas que podem se utilizar do mesmo “Jogo Sério: Uma competição mental, jogada com um computador, de acordo com regras específicas, que utiliza entretenimento para ser usada no futuro por governo ou treinamento corporativo, educação, saúde, políticas públicas, e os objetivos de comunicação estratégica". Zyda (2005) descreve uma experiência do exército americano envolvendo o jogo “*America's Army Game*”, destacando que o mesmo é considerado uma “revolução no pensamento sobre o potencial dos videogames para domínios de não entretenimento”.

Com base nas definições anteriores entende-se que o jogo sério possui uma conotação lúdica para entreter o jogador. Uma atividade lúdica envolve práticas simples que, ao estarem corretamente direcionadas para gerar conhecimento, tornam o aprender algo emocionante e fator desafiador para os alunos.

> “Para fins de treinamento, os jogos sérios são aplicados para simular situações críticas que envolvam algum tipo de risco, tomada de decisões ou, ainda, para desenvolver habilidades específicas. Para fins de ensino-aprendizagem, podem-se simular situações em que o uso de um conhecimento seja necessário para a evolução do jogo” (SOARES, 2012).

Soares (2012) classifica os jogos sérios em três categorias fundamentais: Conscientização (destacam um novo problema), Construção de Conhecimentos (demanda conhecimentos prévios para integrar novos cenários de solução do problema) e Treinamento (alia as categorias anteriores à possibilidade de se realizar repetidas vezes).

Acerca das características de bons jogos, Gee (2005), descreve as seguintes características:

- Interatividade dos conteúdos: Todo conteúdo a ser ensinado deve ser interativo;
- Produção: O jogador deve ajudar a criar o jogo, sendo mais escritor do que leitor, proporcionando sensação de atuação;
- Customização: Um jogo sério deve se adaptar ao jeito de jogar e aprender;
- Consolidação: Após uma nova habilidade ser ensinada ela deve ser consolidada pelo exercício em novos contextos;
- Informação sob demanda: As informações e conteúdos devem aparecer em pequenas doses;
- Sentidos contextualizados: Novos sentidos devem possuir contextos. Palavras seguidas de sua definição não ensinam tão bem quanto sentidos aplicados a uma ação ou diálogo;
- Incentivo à lateralidade: Deve-se incentivar à exploração;
- Ferramentas inteligentes: O jogador deve saber quando e como usar os recursos;
- Performance antes da competência: Fazer antes de ser bom em algo, com o objetivo de desenvolver a proficiência.

Além de todos esses elementos, os jogos devem possuir regras bem definidas. As regras que regem um jogo sério devem estar diretamente relacionadas com os objetivos pedagógicos do jogo. As regras definem a jogabilidade do jogo (ou *gameplay*). O GDD (Game Design Document) é o documento utilizado para definir o jogo em detalhes.

> “Antes de desenvolver um jogo, deve-se produzir um documento de design, que irá conter a descrição do game play, portanto, as regras em detalhes. Este documento de design pode ser encarado como uma espécie de manual de instruções para futuros desenvolvedores. De fato tão importante é esse documento, que o processo de desenvolvimento não pode começar sem que ele esteja pronto.” (FEIJÓ et. al., 2010).

Baseado nas afirmações anteriores pode-se afirmar que jogos sérios têm como principal objetivo a educação ou o treinamento e estão fortemente ligados ao fator lúdico, como forma de prender a atenção do aluno.

Além disso, jogos sérios podem ser aplicados em conjunto com metodologias tradicionais já conhecidas.

> "Em metodologias de ensino tradicionais, a definição formal de algoritmos é apresentada, seguida da resolução de problemas em pseudocódigo e/ou em uma linguagem de programação. Nestas, observa-se que os discentes apresentam dificuldade de abstração, no uso da sintaxe e se mostram desmotivados a prosseguir os estudos." (MELO et. al., 2011, p.1).

É sabido que disciplinas como algoritmos são complexas tanto para serem lecionadas como para serem aprendidas pelos alunos. Neste aspecto, métodos tradicionais de ensino podem ser mais bem difundidos em sala de aula quando empregados em conjunto, com atividades lúdicas, levando consequentemente o aluno a querer aprender.

Já existem alguns esforços no sentido de aproximar a realidade encontrada no dia a dia da Engenharia de Software com as práticas educacionais, é o caso, por exemplo, do LQPS (Laboratório de Qualidade e Pesquisa de Software) da UNIVALI, que em parceria com a instituição implementadora do MPS.Br em empresas, Incremental Tecnologia, desenvolveram alguns jogos com intuito de difundir conceitos de Engenharia de Software, principalmente voltados às práticas dos principais modelos de capacidade da maturidade de processo.

> "Até o momento, três jogos educativos já foram elaborados: (1) A Ilha dos Requisitos; (2) U-TEST e (3) X-MED. Estes jogos fazem uso de aspectos lúdicos, desafios, dicas e rápido feedback ao jogador, com o objetivo de contribuir para a compreensão das práticas destes processos de desenvolvimento de software" (ZOUCAS et. al., 2010).

3. Trabalhos Correlatos

A busca por trabalhos correlatos concentrou-se na pesquisa de artigos científicos que abordassem a problemática da forma de aplicação do ensino de engenharia de software, promovendo uma reflexão acerca de alternativas, principalmente as voltadas ao uso de jogos. Ao todo foram abordados três trabalhos correlatos, que estão listados a seguir.

O primeiro trabalho correlato denominado: "*Abordagem de Apoio ao Ensino e Aprendizagem de Teste de Software Baseada em Jogos Sérios e Mundos Virtuais*", é direcionado para a uma proposta de uma abordagem de apoio ao processo de ensino-aprendizagem focado nas estratégias de teste de software (SILVA et. al., 2013). A proposta trata da criação de um jogo sério focado nas práticas de teste de software, mais especificamente na estratégia de teste. O jogo visa à simular as interações de uma equipe de teste de uma empresa de desenvolvimento de software. O jogo é baseado em fases, o aluno ingressa no jogo com o papel de testador, analista de teste, arquiteto de teste e por fim líder de teste.

O jogo ainda está sendo desenvolvido, os autores pretendem realizar uma análise detalhada dos resultados entre o uso de metodologias tradicionais de ensino e o jogo sério para determinar que essa estratégia é válida podendo auxiliar no processo de ensino-aprendizagem de teste de software.

Já o segundo trabalho é o "*A Game Generator – The Framework for an educational system development game*". Trata-se de um projeto que tem por finalidade a construção de um software capaz de fornecer capacidades de geração de jogo sério, visando práticas educacionais simulando o processo de desenvolvimento de software (LANDRO, 2007). Todo o software gerador de jogos é centrado em uma estrutura de dados ao invés de regras em sistema, sendo assim ao alterar os dados poderá ser alterado o comportamento do jogo. O autor deixa bem claro, ao citar outros trabalhos, que normalmente esses estão concentrados principalmente na parte gráfica do jogo e, na visão dele, para jogos de simulação os gráficos não são a principal preocupação e sim a lógica complexa do jogo.

O terceiro trabalho denominado "*Projeto InVision Framework: Um Framework para suportar a criação e uso de jogos no ensino*". O projeto tem como principal objetivo a criação de um framework que possa ser extensível através de scripts, podendo ser adaptado em várias disciplinas no decorrer do curso, com foco na área de ciência da computação (ALVES, 2011). Um dos

principais argumentos do autor é a baixa procura em cursos de computação e o alto índice de desistência.

Em relação à proposta do gerador, o primeiro trabalho trata da linha de uso de jogo sério, porém o foco é restrito a uma única área e não permite a geração de cenários. O segundo é o que mais se assemelha ao presente trabalho, pois fornece uma ferramenta capaz de prover ao professor a elaboração de jogos criando cenários para serem "jogados" pelos alunos. Um ponto fraco é o controle pobre dos jogadores como controle por equipes. Já o terceiro possui um foco mais arquitetural, no qual se permitiu criar um conjunto de ferramentas capazes de servirem como um framework realmente para que se possa construir um novo jogo utilizando-as, disponibilizando para isto uma estrutura de classes desenvolvidas em .NET.

Nem todos os autores apresentam em seus estudos o resultado prático do uso de jogos sérios no ensino/aprendizagem, mas Zoucas et. al. (2010), abordando a avaliação de três jogos educacionais, relata os resultados obtidos a partir da avaliação de cada jogo utilizando o teste de hipótese não paramétrico de Mann-Whitney, para a comparação do pré-teste e do pós-teste, os resultados constam na Tabela 1:

Jogo	*Avaliação*	*Resultado % de Aprendizado*
A Ilha dos Requisitos	Participaram 31 alunos (16 no grupo experimental e 15 no de controle). Os testes avaliaram a capacidade do jogo em ensinar engenharia de requisitos nos níveis de entendimento e compreensão.	95%
U-Test	Participaram 12 alunos (6 no grupo experimental e 6 no de controle). O teste avaliou a capacidade do jogo em ensinar teste de software nos níveis de entendimento, compreensão e aplicação.	95%
X-MED	Muitos alunos relataram que o jogo os auxiliou a aprender sobre os conceitos e processo de medição. Mas os resultados estatísticos não confirmaram tais afirmações. O fato relatado é que a versão atual do jogo não está adequada para apoiar efetivamente a aprendizagem e desta forma uma nova versão foi desenvolvida, sem avaliação ainda.	-

Tabela 1 – Resultado da avaliação de jogos aplicados na prática (Adaptado de ZOUCAS et. al., 2010).

De fato observa-se, através dos resultados da pesquisa realizada e exibida na Tabela 1, que o uso de jogo sério pode influenciar os alunos ao desafiá-los e através da forma lúdica de expor as situações ele contribui de fato para o aprendizado.

Existe a necessidade de se voltar atenção para o objetivo do presente trabalho, que é propor uma ferramenta para geração de jogos sérios em alto nível, voltados principalmente para as disciplinas ligadas à engenharia de software. Desta forma é importante o entendimento e contextualização dos trabalhos correlatos, para que se entenda onde estão as necessidades, dificuldades e similaridades entre as propostas e os diferenciais deste.

4. Estudo de Caso

O jogo "A Ilha dos Requisitos" definido no trabalho denominado "*Promovendo a Aprendizagem de Engenharia de Requisitos de Software Através de um Jogo Educativo*", foi utilizado como estudo de caso de um jogo que pode ser criado pelo gerador simulando o seu funcionamento, através do mapeamento completo do jogo.

Este é um "jogo desenvolvido em meio digital que faz uso de aspectos lúdicos e de desafios para promover uma aprendizagem mais natural do processo de Engenharia de Requisitos" (THIRY et. al., 2010). No trabalho os autores explanam sobre o jogo, incluindo a estória envolvida, a forma de jogo, os objetivos, diferenciais em relação à outros correlatos do trabalho em questão, a classificação do jogo, a forma de interação e demais aspectos.

O jogo está classificado nos gêneros de "Aventura" e "Estratégia". A classificação do jogo quanto ao gênero "Aventura", se deve ao fato da "Ilha dos Requisitos" ser um lugar desconhecido, e o avanço depende da resolução de desafios. Já o gênero "Estratégia" justifica-se, pois durante o jogo o jogador deve especificar os requisitos para construir uma determinada solução.

Mas para que fosse possível mapear completamente o jogo "A Ilha dos Requisitos", realizou-se contato via e-mail com um dos autores, onde fora possível acessar o jogo através de um link. Desta forma verificou-se com maior riqueza de detalhes como o jogo funciona, qual linha de tempo da execução do jogo enquanto jogador final, quais as fases existentes no jogo incluindo a jogabilidade de cada uma delas através da observação de aspectos como: tema de cada fase e objetivos, quantidade de dicas e forma de uso e disposição visual de objetos.

Como se pode observar na Figura 1, o jogo conta com um contexto lúdico que chama a atenção.

Figura 1: Tela Inicial do jogo "A Ilha dos Requisitos" (Site Oficial do Jogo)

Já na Figura 2, o jogo apresenta uma área de identificação do jogador e uma lista com as regras do jogo.

Figura 2: Tela de Identificação do Jogador e Regras (Site Oficial do Jogo)

A partir deste acesso e experimentação do jogo, foi possível a elaboração do GDD disponível no Apêndice B, bem como a confecção de um protótipo de baixa fidelidade em papel descrevendo como seria o aspecto do jogo após modelado no gerador.

5. Fases para Construção de Jogos no Gerador

Como forma de demonstrar visualmente as atividades necessárias para a criação dos jogos pelo professor, condução das atividades com os alunos e comunicação, foi elaborado um processo completo em formato de fluxograma. Este processo será necessário para identificar todos os artefatos envolvidos na definição do gerador a ser projetado.

Além disso, o fluxograma ajuda a definir uma linha de atividades a serem conduzidas pelo professor e também pelos alunos para a elaboração e execução dos jogos. Por se tratar de um processo extenso, o mesmo foi particionado conforme a Tabela 2:

Fase	*Sub-Fase*	*Resumo*
Inicialização		O executor é o professor e basicamente ele irá neste momento definir os dados básicos do jogo.
	Criar Cenário Principal	Procedimento onde o cenário principal do jogo é criado para que posteriormente as fases sejam definidas
	Definir as Fases	Nesta atividade o professor deverá elaborar todas as fases do jogo, conforme definição contida no GDD.
Participantes		Neste momento o professor poderá optar por realizar o jogo e pré-definir os participantes ou então apenas iniciar o jogo.
Execução		A execução é quando o jogo é liberado para que os participantes possam jogá-lo. A partir deste ponto as atividades de gerenciamento e controle são disparadas e realizadas pelo professor.
	Gerenciar Execução	Engloba as atividades de gerenciamento da execução dos jogos pelo.
	Monitoração e Controle	A monitoração é realizada também pelo professor e poderá resultar em intervenções durante o jogo como o envio de mensagens aos participantes.
Coletar Resultados		Esta atividade resultará em observações coletadas pelo professor e vinculadas ao jogo para que posteriormente possam ser revisadas.

Tabela 2 – Organização das fases da construção de jogos no gerador (Acervo do Autor)

No Apêndice A estão as definições detalhadas acerca das fases para a construção de jogos utilizando o gerador.

Através da definição detalhada do processo, fora possível identificar a maioria dos recursos que o gerador de jogo sério deverá dispor. O complemento dos recursos do gerador é alcançado com os requisitos, casos de uso e o GDD, presentes na seção 6.

6. O Projeto do Gerador de Jogos

O uso de protótipos de baixa fidelidade foi importante para assegurar que o mesmo funcionará em ambiente real de uso. Os protótipos foram divididos em três grupos: (1) Protótipo de baixa fidelidade wireframe em papel para montar o jogo "A Ilha dos Requisitos", utilizando o processo definido na seção 4; (2) Protótipo de baixa fidelidade wireframe em ferramenta de prototipação (*Pencil*) para principais interfaces do gerenciador do gerador; e (3) Protótipo de baixa fidelidade wireframe para a ferramenta de desenho.

Os protótipos são uma importante ferramenta em situações onde não se tem uma clareza de detalhes logo de início, sendo o protótipo do tipo baixa fidelidade útil, pois a confecção é rápida e pode-se fazê-lo utilizando apenas rabiscos. Como pode ser observado na Figura 3, para o protótipo do jogo "A Ilha dos Requisitos", foram mapeados vários itens e componentes necessários.

A opção do jogo "A Ilha dos Requisitos" para ser mapeado no GDD (conforme Apêndice B) e este servindo como base para a definição das funcionalidades do gerador, deve-se (1) ao fato do jogo já possuir um enredo artístico definido; (2) uma apresentação lúdica bem divertida; (3) por estar no contexto de engenharia de software; (4) ser testado em aplicação real conforme pode ser visto na seção 2; e (5) por ser foco central em artigo correlato, apresentado na seção 3.

Figura 3 – Artefatos do protótipo wireframe de baixa fidelidade em papel (Acervo do Autor).

A execução do protótipo de baixa fidelidade em papel foi realizada e filmada, seu conteúdo foi postado no Youtube e está disponível através do link: *<http://youtu.be/lyOgVkZ4AKo>*.

Já os protótipos de baixa fidelidade feitos através da ferramenta de prototipação (*Pencil*), possuem um nível de detalhamento maior e ajudam a definir com mais precisão (em relação ao papel), as características do gerador. As principais interfaces foram mapeadas por protótipos e dentre elas as que merecem destaque estão listadas a seguir.

Na Figura 4, é demonstrado um protótipo da tela de login, onde tanto professores como alunos podem iniciar o uso do gerador (no caso de professores) ou de jogos (no caso apenas de alunos).

Figura 4 – Protótipo wireframe de baixa fidelidade da tela de login (Acervo do Autor).

Na Figura 5, está a tela inicial do gerador para o professor, onde ele poderá gerenciar os seus jogos, criar novos jogos, definir turmas de alunos e alunos bem como gerenciar os papéis que poderão ser exercidos posteriormente pelos alunos.

Figura 5 – Protótipo wireframe de baixa fidelidade do menu principal (Acervo do Autor).

Importante frisar que os jogos que serão mostrados em “Exibindo Jogos”, são aqueles definidos pelo professor. Conforme a Figura 5, que faz parte do menu principal, o professor poderá manipular os seus jogos conforme necessidade. Existem também disponíveis as ações: Meu Perfil e Logoff.

Na Figura 6, é exibido protótipo da tela onde são definidos os dados do jogo quando um é criado ou manipulado pelo professor. Os primeiros detalhes do jogo, como: Nome, Objetivos, Data Limite para disponibilização e Logo, são definidos em Dados Básicos.

Figura 6 – Protótipo wireframe de baixa fidelidade dos dados básicos do jogo (Acervo do Autor).

Na Figura 7, é exibido o protótipo da tela de definições do jogo na aba “Nível e Modo de Jogo”, onde os detalhes a respeito do nível de dificuldade, pré-requisitos, modo de jogo são definidos. Além disso pode-se configurar se os participantes são pré-definidos, se o jogo será em Equipes, se é contra o tempo e se é permitido o envio de mensagens entre os participantes do jogo.

Figura 7 – Protótipo wireframe de baixa fidelidade da definição do nível e modo de jogo (Acervo do Autor).

Na Figura 8 é exibido o protótipo da tela de definições do jogo na aba “Baú de Recompensas”. Nesta etapa será necessário definir quais são os itens de recompensa que o jogador terá durante o jogo, sendo que alguns deles como é o caso do item 4 (Bateria) o jogador deverá encontrar escondido no jogo.

Figura 8 – Protótipo wireframe de baixa fidelidade da definição do baú de recompensas (Acervo do Autor).

A Figura 9 apresenta o protótipo da tela onde o professor deverá configurar os cenários do jogo.

Figura 9 – Protótipo wireframe de baixa fidelidade da configuração de cenários (Acervo do Autor).

Como pode ser observado na Figura 9 para cada cenário uma descrição e imagem de fundo deverá ser definida. A opção "Disposição das Fases" indica a forma como o gerador irá exibir as fases ao jogador, que pode ser automática ou desenhada onde o professor irá utilizar a ferramenta de desenho (Figura 13) para dispor as fases no cenário.

Além disso, existe ainda a necessidade de configurar os eventos, componentes (utilizados posteriormente pelas fases do cenário) e as fases (que podem ser configuradas a partir desta tela também).

Na Figura 10, é apresentado o protótipo da tela onde será possível a realizar a configuração de cada uma das fases do cenário.

Figura 10 – Protótipo wireframe de baixa fidelidade da configuração das fases (Acervo do Autor).

Para cada fase deve-se informar o seu nome, descrição, imagem de fundo (que será um componente do cenário da fase), tipo da fase. Além disso pode-se configurar se a mesma será obrigatória, bloqueadora (não permite avançar sem passar por ela), se pode voltar ao cenário principal e se permite reiniciar (sendo necessário neste caso informar o número de reinícios).

Existe ainda a necessidade de configurar os eventos para a fase (que pode ser visto na Figura 11). As configurações específicas são aplicadas dependendo do tipo da fase por exemplo: se a fase for do tipo "Organizar Arranjo de Objetos na Sequencia Correta", então deve-se configurar quais são os objetos e qual a sequencia correta de arranjo esperada para a fase ser concluída com sucesso. Como pode ser observado no diagrama de classes do Apêndice C, fora definido um estrutura de classes para prever esta situação e facilitar a manutenção futura do gerador quando novos tipos de fase forem adicionados ao mesmo.

Os tipos de fase disponíveis até o momento (definidos juntamente com a proposta do gerador) são: (1) Organizar Arranjo de Objetos; (2) Fase Bônus; (3) Seleção de Opções Corretas (selecionar na lista); (4) Classificar objetos; (5) Mercado de Coisas; e (6) Apenas Visitar. O comportamento de cada uma delas é definido através da configuração de eventos definido na Figura 11.

Na Figura 11 é exibido o protótipo da tela de configuração de eventos do cenário ou das fases. Para um cenário ou uma fase vários eventos podem ser definidos e para cada evento uma série ordenada de ações podem ser realizadas.

Figura 11 – Protótipo wireframe de baixa fidelidade da tela de configuração de Eventos (Acervo do Autor).

Na Figura 12, estão dispostos os protótipos das telas de manutenção de Alunos e Turmas.

Figura 12 – Protótipo wireframe de baixa fidelidade da tela de configuração de Eventos (Acervo do Autor).

Nas telas apresentadas na Figura 12, o professor poderá gerenciar os alunos incluindo-os manualmente ou então apenas mantendo aqueles cadastrados. A ideia é permitir que o professor possa visualizar os diversos participantes (alunos) e seus jogos, além disso, poderá também manter as turmas de alunos vinculando-os nas turmas.

O professor poderá ainda enviar mensagens aos alunos individualmente ou coletivamente para através das turmas.

Na Figura 13, está definida através de protótipo wireframe de baixa fidelidade a ferramenta de desenho, que poderá ser utilizada pelo professor para manipular visualmente através de uma ferramenta de alto nível, o comportamento do cenário principal e das fases.
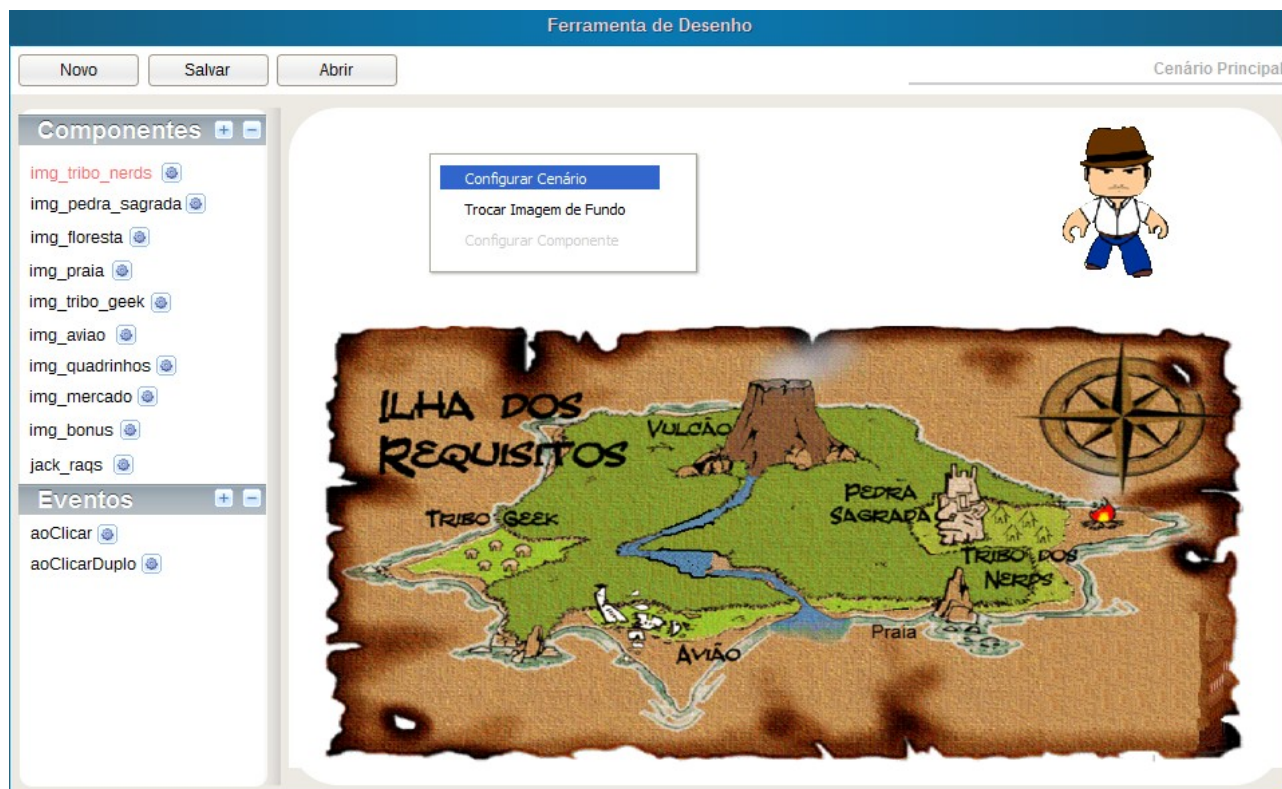
Figura 13 – Protótipo wireframe de baixa fidelidade da ferramenta de desenho (Acervo do Autor).

A ferramenta de desenho exibirá os componentes de acordo com o cenário ou fase a desenhar (quando eles estiverem disponíveis). Mas o professor poderá adicionar novos componentes diretamente na ferramenta de desenho, bem como acessar a configuração do comportamento dos eventos na área de eventos, onde serão exibidos os eventos específicos do componente selecionado.

Através da confecção dos protótipos foi possível uma ligação clara entre o processo definido e a forma como o professor e aluno farão a interação através do gerador de jogo sério. Um documento acaba complementando o outro e desta forma obtém-se uma ótima fidelidade em relação ao gerador executável final.

As fases para a construção de jogos (definido na seção 4) e os protótipos apresentados no início desta seção são realmente muito importantes, mas o projeto do gerador de jogo sério somente estará completo com a definição do projeto arquitetural do software (através dos diagramas da UML) e a definição do GDD.

Para o projeto do software, foram utilizados os principais diagramas da UML através da ferramenta Enterprise Architect (SPARKX, 2011), como o diagrama de casos de uso. A realização dos principais casos de uso e diagrama de classes de domínio, que estão disponíveis no Apêndice C. Além disso, serão explicitadas as regras de negócio e os requisitos funcionais.

Conforme pode ser observado na Tabela 3, estão dispostas as regras de negócio.

ID	*Descrição*
RN01	O Gerador de Jogos é destinado ao professor. Ele é quem será o administrador do sistema. O Professor será o administrador do gerador de jogos. A administração do gerador consiste em definir turmas de alunos, definir alunos, elaborar jogos, etc.
RN02	O jogador do jogo é o aluno e deverá cadastrar-se para poder jogar
RN03	Jogo sério é um software ou hardware que tem como objetivo o foco no aprendizado
RN04	Em caso de jogos contra o tempo, o mesmo somente poderá ser computado quando estiver na fase, sem a exibição de mensagens (ex.: popup)
RN05	Ao acessar a área de administração do gerador, o professor poderá selecionar um dos jogos por vez para manutenção.
RN06	Baú de Recompensas, é um local onde ficam armazenadas as recompensas coletadas pelo aluno durante a execução de um jogo.
RN07	Recompensa são objetos de diversos tipos e formas que é coletado pelo aluno durante a execução do jogo, através de atividades completadas com sucesso. Cada objeto de recompensa prevê um determinado número de pontos
RN08	Artefato de valor é algo que ao iniciar o jogo estará disponível ao jogador e poderá ser utilizado no decorrer do jogo. Cada artefato de valor deverá ter uma pontuação vinculada
RN09	Cada jogo criado deverá estar ligado aos objetivos pedagógicos do assunto e da disciplina.
RN10	A única forma de comunicação entre professor e jogadores durante os jogos deverá ser via mensagem interna.
RN11	Caso o jogo tenha participante pré-definido então ele será um “jogo fechado”, caso contrário será um “jogo aberto”.

Tabela 3 – Descrição das regras de negócio (Acervo do Autor).

Na Tabela 4, estão definidos os requisitos funcionais do gerador.

ID	*Descrição*
RF01	Permitir que o professor mantenha as turmas de alunos.
RF02	Permitir que o professor mantenha os alunos e o relacionamento entre eles e as turmas.
RF03	Permitir que o professor cadastre/altere e exclua os jogos e seus dados básicos.
RF04	Permitir que o professor libere e bloqueie o jogo para que fique disponível para execução.
RF05	Permitir que seja configurado o baú de recompensas para o jogo criado.
RF06	Permitir que seja definida a lista de regras do jogo
RF07	Permitir que sejam definidos os cenários do jogo
RF08	Permitir que sejam definidas as fases do cenário
RF09	Permitir que o professor adicione participantes ao jogo.
RF10	Permitir que o professor crie equipes para disputa do jogo. Os membros das equipes são os participantes
RF11	Permitir que o professor possa se comunicar com os participantes através de mensagem interna
RF12	Permitir que o professor libere e bloqueie apenas uma fase do jogo para execução por vez
RF13	Deverá exibir uma área contendo os componentes disponíveis seja para o cenário ou para a fase
RF14	Deverá permitir acesso à tela de configuração do cenário a partir da área de desenho
RF15	Deverá exibir uma área contendo os eventos de cada componente selecionado.
RF16	A área de desenho deverá receber os objetos constantes na lista de componentes através da opção de arrastar e soltar.
RF17	Deverá permitir que o desenho (cenário / fase) seja salvo a qualquer momento em formato próprio.
RF18	Deverá permitir que o professor carregue desenhos salvos anteriormente e vincule ao jogo atual.
RF19	Permitir que professor e aluno cadastrem-se na página principal do gerador
RF20	Permitir que o aluno possa realizar login como jogador.
RF21	Deverá exibir uma lista com os jogos disponíveis para o aluno.
RF22	Deverá ser possível o aluno salvar o estado atual do jogo para poder continuar posteriormente.
RF23	Deverá ser possível nos cenários que o aluno acesse as regras do jogo, o baú de compensas, o seu perfil, caixa de mensagens e também finalize aquele jogo.
RF24	Permitir que o aluno cadastre-se para poder iniciar o ambiente de jogos
RF25	Deverá disponibilizar uma forma de o aluno consultar as mensagens do jogo: caixa de entrada.
RF26	Deverá ser possível manter uma lista genérica de papeis que serão utilizados em diferentes momentos nos jogos.

Tabela 4 – Descrição dos requisitos funcionais (Acervo do Autor).

As regras de negócio constantes na Tabela 3, são realizadas através de um ou vários requisitos funcionais listados na Tabela 4. Cada requisito funcional poderá ser realizado por um ou mais casos de uso, bem com um caso de uso poderá realizar mais de um requisito funcional. Os casos de uso mapeados estão dispostos na Figura 14, Figura 15, Figura 16 e Figura 17, pois foram agrupados pelos pacotes: (1) Administração Básica; (2) Elaboração de Jogos; (3) Execução de Jogos e (4) Gerenciamento de Jogos.
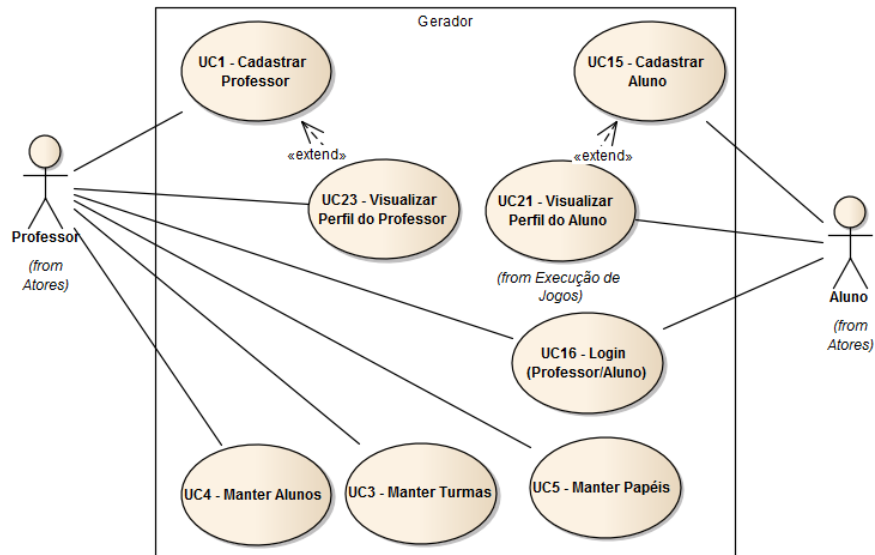
Figura 14 – Diagrama de Caso de uso Administração Básica (Acervo do Autor).

Na Figura 15, está sendo exibido o Diagrama de Caso de Uso de elaboração dos jogos. Ele contempla todas as atividades necessárias para que o professor possa criar e manter os jogos.
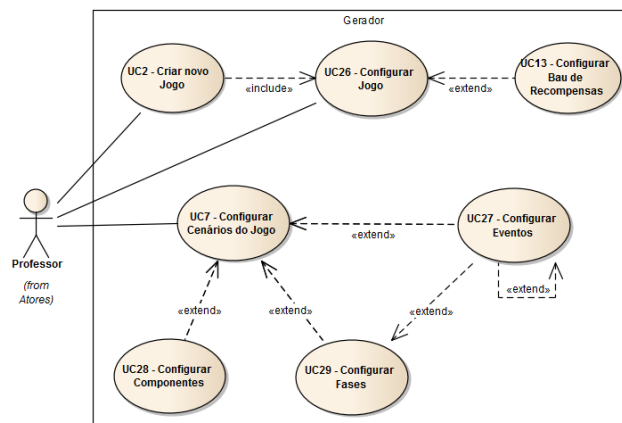
Figura 15 – Diagrama de Caso de uso Elaboração dos Jogos (Acervo do Autor).

Na Figura 16, é exibido o Diagrama de Caso de Uso para Execução de Jogos. Nele estão dispostos todos os casos de uso que o aluno poderá realizar para poder executar os jogos.
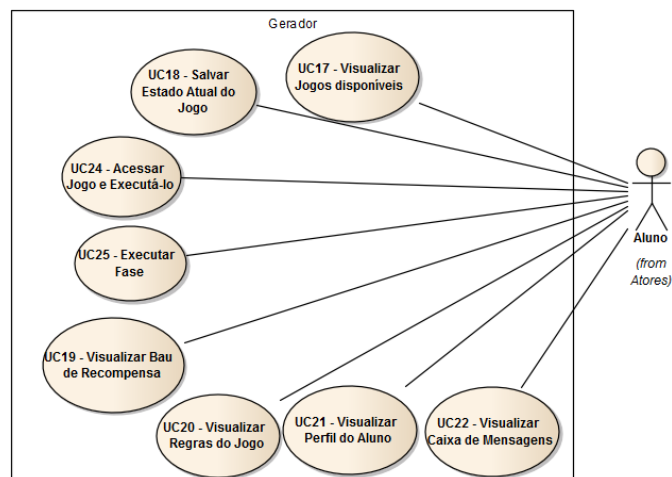
Figura 16 – Diagrama de Caso de uso Execução dos Jogos (Acervo do Autor).

Na Figura 17 é exibido o diagrama de casos de uso do pacote de gerenciamento de jogos. Nele estão os casos de uso que podem ser utilizados pelo professor para realizar as atividades de gerenciamento enquanto o jogo está em andamento.
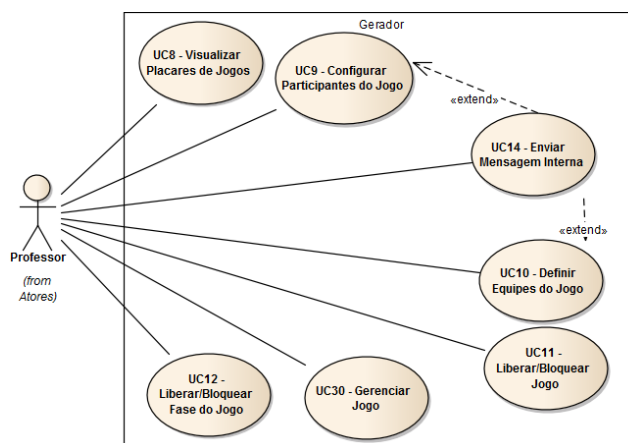
Figura 17 – Diagrama de Caso de uso Gerenciamento dos Jogos (Acervo do Autor).

Além disso, é necessária a elaboração de um *template* próprio de *Game Design Document* (GDD). Este template será utilizado como um guia para a elaboração dos jogos pelo professor. "Como tal, todos os detalhes necessários para construir o jogo deve ser abordado no documento (ou documentos de suporte). Se não estiver no documento, então ele provavelmente não vai estar no jogo." (BALDWIN, 2005).

Como forma de evidenciar a usabilidade do gerador proposto, foi elaborado o GDD para o jogo "A Ilha dos Requisitos", definido por THIRY et. al. (2010) e a partir dele fora definida uma parte dos requisitos funcionais do gerador, ele pode ser visualizado no Apêndice B.

7. A Ilha dos Requisitos no Gerador de Jogo Sério

Como alternativa ao mapeamento de jogos sem que fosse realizado o devido uso para avaliar sua capacidade de transmissão de conhecimento, fora optado por utilizar o jogo "A Ilha dos Requisitos" (THIRY et. al., 2010), sendo que foi realizada a construção do mesmo utilizando o gerador de jogo sério.

A construção foi realizada através do protótipo wireframe de baixa fidelidade em papel, conforme definido na seção 6 e também conforme definição completa contida no GDD. A passagem do jogo para o gerador de jogo sério, implicou na modificação do comportamento de alguns fatores em relação ao jogo original como, (1) o acesso ao jogo; (2) a visualização dos itens

no cenário principal (por exemplo, no jogo original existe a opção “Escore”, já na proposta do gerador esta opção está disponível através do acesso ao “Baú de Recompensa”); (3) as regras de pontuação são automaticamente exibidas no baú de recompensas; (4) os textos no início de cada fase e as dicas serão exibidos utilizando o recurso de popup; (5) não será necessário o aluno acionar a opção de avaliação das respostas da fase, pois o jogo fará a avaliação automaticamente quando a última informação for definida, este recurso dependerá do tipo da fase; (6) nas fases onde existe um conjunto de respostas e nas fases de organização de imagens na sequencia correta a disposição dos objetos ao iniciar será aleatório e não estático como é no jogo original; (7) a lista regras do jogo ficará disponível através de opção sempre visível ao jogador no cenário principal e nas fases.

Além do comportamento diferenciado do jogo, foram adicionadas mais duas fases ao mesmo, respeitando os objetivos pedagógicos do jogo e aumentando o lado lúdico. Trata-se de duas fases bônus: (1) fase 4, repetir a sequência (esta fase bônus disponibilizará ao jogador a opção de repetir a sequencia de cores para ganhar mais pontos no seu baú de recompensa), esta fase somente estará disponível quando o jogador completar a fase 3; (2) fase 8, mercado (o jogador poderá comprar uma coisa no mercado caso ele tenha encontrado a bateria do notebook de Jack no jogo). Neste caso o mercado trata-se da compra de ferramenta de gestão de requisitos.

8. Considerações Finais

Este trabalho consistiu na formação de uma proposta para um gerador de jogo sério, voltado principalmente para que o professor possa elaborar jogos educacionais em alto nível e gerenciá-los. Para que a proposta fosse concretizada uma revisão teórica foi realizada entre o estudo de jogo sério e trabalhos correlatos.

Durante a pesquisa realizada acerca de assuntos correlatos, percebeu-se que a maioria dos trabalhos está ligada a um foco específico de conhecimento relacionado às disciplinas de computação, como gerência de projetos, teste de software, algoritmos, etc. e nenhum dos encontrados permitem um gerenciamento mais detalhado dos membros das equipes participantes. A presente proposta de gerador de jogos disponibiliza ao professor uma ferramenta que permite ao mesmo elaborar seus jogos conforme demanda em alto nível (sem necessidade de conhecimento de programação) e gerenciamento avançado dos jogos disponíveis e em execução bem como dos participantes dos jogos através da manutenção de alunos e turmas.

Para a elaboração da proposta foi necessária a construção de um processo contemplando todas as atividades necessárias para o (1) gerenciamento básico do gerador; (2) elaboração de jogos; e (3) execução de jogos incluindo o gerenciamento da execução. A partir disso trabalhou-se na elaboração dos protótipos de baixa fidelidade em conjunto com a elaboração do GDD. Desta forma a elaboração do projeto do gerador foi muito mais efetiva, pois a maioria dos artefatos já estavam levantados e conectados, praticamente definidos.

O mapeamento do jogo “A Ilha dos Requisitos” foi importante, pois independente do jogo que fosse abordado para o mapeamento, pôde-se montar o GDD, o que auxiliou bastante nas etapas seguintes. O grande problema na elaboração de um jogo é a montagem de um “Enredo” que prenda a atenção do jogador, que contenha a atração do lúdico e principalmente tenha uma atenção especial ao conteúdo a ser aprendido pelo aluno. Por isso, antes de iniciar a montagem do jogo no gerador, será sempre recomendado a utilização do GDD para a completa definição do mesmo. Ao mapear o jogo, novas fases foram adicionadas e o comportamento de algumas partes já existentes foi modificado, demonstrando assim o potencial de inovação do gerador.

Através do protótipo wireframe de baixa fidelidade em papel, foi possível simular o acesso ao jogo, os quadrinhos (que exibem a estória do jogo), o cenário principal (onde se encontra o mapa), o baú de recompensas, as opções disponíveis no cenário principal e interface padrão das fases e a fase 1.

Observa-se que com um gerador desse tipo de jogo é possível fornecer ao professor uma ferramenta que deverá ser aliada nas atividades já existentes em sala de aula. Neste sentido o

próximo passo do trabalho será o desenvolvimento do gerador de jogo sério e consequentemente a adição de novos componentes ao gerador como modos de jogo, tipos de fase, tipos de eventos, tipos de ação, níveis de dificuldade e itens de recompensa. Além disso, novos gêneros de jogos podem ser disponibilizados além dos gêneros aventura e estratégia presentes no jogo do estudo de caso.

Referências

ALVES, Daniel Ferreira Monteiro. **Projeto Invision Framework:** Um Framework para suportar a criação de jogos no ensino. 2011. 108. Grau: Dissertação – Universidade Federal de Goiás, Goiânia, 2011.

BALDWIN, Mark. **Game Design Document Outline**. Versão: 0.1. Local: Site do Autor, 10/10/2005. Disponível em: <http://www-personal.engin.umd.umich.edu/~bmaxim/cis488/BaldwinGameDesignDocumentTemplate.doc.>. Acesso em: 04 jun. 2013, 20:59.

FEIJÓ, Bruno. CLUA, Esteban. DA SILVA, Flávio S. Corrêa. **Introdução à Ciência da Computação com Jogos**: Aprendendo a programar com entretenimento. Primeira Edição. Rio de Janeiro: Elsevier, 2010, 288.

GEE, James Paul. **Good Video Games Plus Good Learning**: Collected Essays on Video Games, Learning and Literacy. New York: Peter Lang Publishing, Inc. 2007, 194.

LANDRO, Anne C. **A Game Generator – The Framework for an Educational System Development Game** Tese de Mestrado da Universidade de Oslo, EUA. Disponível em: <https://www.duo.uio.no/bitstream/handle/10852/9769/landro.pdf?sequence=2> . Acesso em 31 mai. 2013, 22:28.

MELO, Jeane C. B.. DA SILVA, Amanda S. C.. DA SILVA, Tatyane S. C.. Ensino de Algoritmos a Nível Médio Utilizando Música e Robótica: Uma Abordagem Lúdica. In: Congresso da Sociedade Brasileira de Computação, 31, 2011, Natal-RN. **Anais Eletrônicos**... Natal: 2011. Disponível em: <http://www.dimap.ufrn.br/csbc2011/anais/eventos/contents/WEI/Wei_Secao_6_Artigo_4_Silva.pdf>. Acesso em: 04 jun. 2013

SILVA, Tarcila G.. BERNARDI, Giliane. MÜLLER, Felipe M.. Abordagem de Apoio ao Ensino e Aprendizagem de Teste de Software Baseada em Jogos Sérios e Mundos Virtuais. In: **XXII SBIE**, 538., 2011, Aracaju. Anais. Aracaju: 2011.

SPARKX Enterprise Architect. Versão 9.0.: Sparkx, 2011.

SOARES, Michele. **Jogos Sérios e Educação**. 2012. 6. Tese – PUC-Rio, Rio de Janeiro, 2012.

TORI, Romero. **Educação sem Distância**: As tecnologias interativas na redução de distâncias em ensino e aprendizagem. Segunda Edição. São Paulo: Editora Senac, 2010, 257.

THIRY, Marcello. ZOUCAS, Alessandra. GONÇALVES, Rafael Queiroz. **Promovendo a Aprendizagem de Engenharia de Requisitos de Software Através de um Jogo Educativo**. Artigo publicado no SBIE, 2010, em João Pessoa, Brasil. Disponível em: < http://www.br-ie.org/pub/index.php/sbie/article/download/1514/1279>. Acesso em 02/ jan. 2014, 19:05.

ZOUCAS, Alessandra. THIRY, Marcello. SALVIANO, Clenio. **Aplicação de Jogos Educativos para Aprendizagem em Melhoria de Processo e Engenharia de Softeware.** Artigo publicado no WAMPS, 2010, em Campinas-SC, Brasil. Disponível em: < http://www.softex.br/wp-content/uploads/2013/11/Wamps-2010_pag-simples.pdf>. Acesso em 12/ jan. 2014, 09:05.

ZYDA, Michael. **From Visual Simulation to Virtual Reality to Games.** Artigo publicado pelo IEEE Computer Society, 2005, em Lima, Perú. Disponível em: <http://e-quipu.pe/upn/wp-content/uploads/2013/05/games.pdf>. Acesso em 15/abr. 2014, 21:48.

WANGENHEIM, C. G.; SILVA, D. A. Qual conhecimento de engenharia de software é importante para um profissional de software? In: **FÓRUM DE EDUCAÇÃO EM ENGENHARIA DE SOFTWARE**, 2009, Fortaleza. Anais. Fortaleza: UFC.

APÊNDICE A – Detalhamento das Fases para a construção de jogos no Gerador

Este apêndice demonstra detalhadamente quais são as fases necessárias para que o professor possa elaborar os seus jogos e os alunos possam jogá-los, sendo que elas estão divididas em: (1) Processo principal do gerador de jogo sério; (2) Criar o cenário principal; (3) Definir as fases; (4) Gerenciar execução e (5) Execução do jogo e Execução da fase.

O processo principal foi organizado ainda nas fases de: (1) Inicialização conforme definido na Figura 18, que mostra as atividades necessárias a serem realizadas para a definição do jogo; (2) Participantes; (3) Execução e (4) Coletar Resultados.
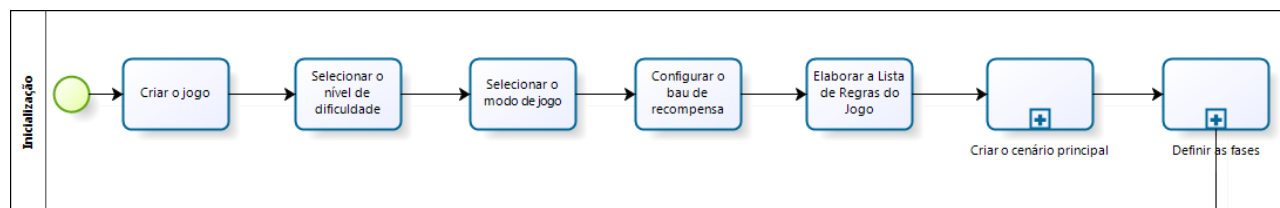
Figura 18: Fase de Inicialização do Processo Principal do Gerador de Jogo Sério (Acervo do Autor).

O executor dessas atividades é o professor. As atividades "Criar o cenário principal" e "Definir as fases", são sub-processos definidos em detalhes posteriormente.

A definição de cada atividade da fase de inicialização é demonstrada na Tabela 5.

Atividade	*Descrição*
Criar o Jogo	O professor deverá definir o jogo, incluindo uma documentação completa de objetivos, forma de aprendizado, referências bibliográficas, etc.
Selecionar o Nível de Dificuldade	Logo após criado o jogo, o professor deverá selecionar o nível de dificuldade a ser aplicado. Isto poderá ser útil para que decisões do jogo sejam tomadas de acordo com a disciplina bem como o conhecimento da turma. Os níveis de dificuldade podem ser Iniciante, Intermediário ou Profissional.
Selecionar o Modo de Jogo	Basicamente esta atividade consiste em o professor selecionar a forma como o jogo será conduzido. As formas podem ser: - **Somente Tabuleiro** - ações sequenciais uma após a outra; - **Quiz** - neste modo as ações são vencidas a medida que respostas são dadas pelo aluno a uma série de questões no contexto da ação. - **Rally** (contra o tempo) - ações com tempos pré-definidos são organizadas e o aluno deverá executá-las no menor tempo. Perdas e ganhos de tempo podem ocorrer. - **Fases sequenciais** - devem ser executadas através de interações definidas.
Configurar o Baú de Recompensas	O Professor deverá configurar o baú de recompensa inicial do jogo (um local onde ficam armazenadas as recompensas ao iniciar o jogo e também as coletadas pelo jogador durante sua execução). Adicionar componentes iniciais, ex.: TIPO INTEIRO, DICA, TEMPO, RECOMPENSA, DADO DA SORTE, FASE BÔNUS. Outras formas de item de recompensa podem ser definidas conforme demanda, através da modificação do comportamento do gerador. Definir pontuação limite para exclusão, significa dizer que quando o aluno chegar a uma determinada pontuação mínima ele deverá ser excluído do jogo. Definir uma pontuação inicial para o jogador.
Elaborar a Lista de Regras do Jogo	O Professor deverá descrever uma lista com as regras do jogo.
Criar o Cenário Principal	O professor criará um cenário. O cenário principal é "o menu do jogo". Ele poderá (ou não) ter um comportamento a ser configurado. Ver mais detalhes na Tabela 3.
Definir as Fases	O professor deverá definir a quantidade de fases que serão executadas no cenário. Uma fase deverá incluir ações a serem executadas para um modo distinto de jogo. Um cenário poderá possuir várias fases. Neste momento o professor deverá definir todas as fases do jogo, criando-as e definindo os dados básicos de cada fase como: tempo limite.

Tabela 5 – Descrição das atividades da fase de Inicialização (Acervo do Autor).

Os sub-processos de Criar Cenário Principal e Definir as fases podem ser visualizados na Figura 19 e Figura 20, respectivamente.
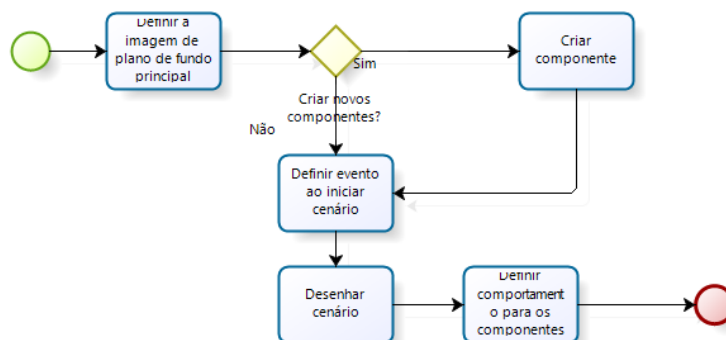
Figura 19 – Descrição das atividades do sub-processo Criar Cenário Principal (Acervo do Autor).

O sub-processo Criar Cenário Principal, será executado para que seja possível o professor definir visualmente como o jogador (aluno) fará a interação com o jogo. Na Tabela 6 está a definição de cada atividade do sub-processo.

Atividade	*Descrição*
Definir a Imagem de Plano de Fundo Principal	Cada cenário poderá ter uma imagem estática de fundo, por exemplo, um mapa, um tabuleiro, etc.
Criar Componente	O professor poderá optar ou não por criar componentes que podem ser imagens (como indicações de onde as fases estão em um mapa). Os componentes são criados no cenário a partir de tipos pré-definidos e podem ser utilizados também nas fases do cenário, como é o caso do img_mercado (definido no GDD, ver Apêndice A) que é utilizado na fase bônus mercado do jogo "A Ilha dos Requisitos".
Definir evento ao iniciar cenário	Os eventos são disparados em momentos pré-definidos. O gerador estará preparado e disponibilizará ao professor uma série de eventos dependendo do contexto. Um evento sempre contará com uma lista de ações que podem ser configuradas para o mesmo, como exibir uma mensagem, iniciar uma fase, desbloquear uma fase, etc. No caso do cenário o evento *aoIniciarCenario()* será executado sempre que for definido. É importante para exibir informações a respeito do jogo para o jogador, exibir um popup com alguma mensagem, exibir uma imagem como um artefato a ser localizado, etc.
Desenhar o Cenário	O desenho do cenário será realizado sobre a o plano de fundo definido para o mesmo. O desenho poderá contar com componentes visuais (como botões e imagens) a serem posicionados sobre a imagem. O professor poderá utilizar a ferramenta de desenho em alto nível para posicionar os componentes no cenário.
Definir comportamento dos componentes	O comportamento dos componentes deverá ser definido. Cada componente terá os eventos *aoClicar()* e *aoClicarDuplo()* disponíveis para definição. Eles poderão, por exemplo, exibir informações de uma fase (popup), acessar a fase, etc. Este comportamento poderá ser alterado pelo professor em outras fases do jogo, por exemplo disponibilizar o mercado de coisas (um tipo de fase Bônus).

Tabela 6 – Descrição das atividades do sub-processo Criar Cenário Principal (Acervo do Autor).

Na definição das fases, conforme Figura 20, o professor deverá definir detalhadamente o comportamento de cada fase dependendo do tipo selecionado. O gerador disponibilizará uma série de tipos pré-definidos, que poderão ser ampliados no futuro.
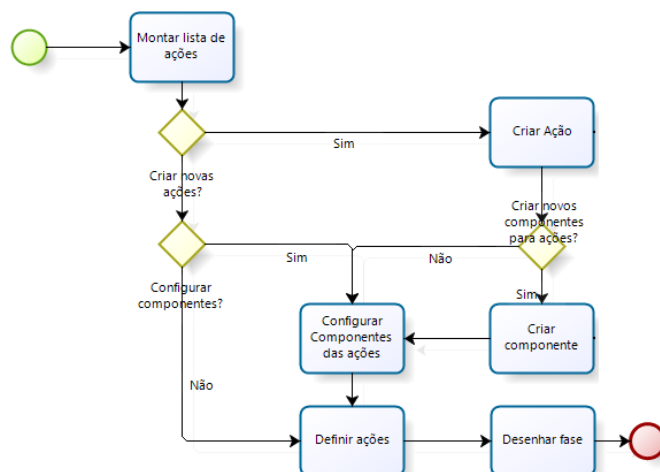
Figura 20 – Descrição das atividades do sub-processo Definir as Fases (Acervo do Autor)

O sub-processo de definição de fases será provavelmente o que consumirá a maior parte do tempo do professor ao criar os seus jogos. A definição do comportamento de cada fase deve ser bem elaborada para que tenha sempre o lúdico como ponto chave e, logicamente, não esquecendo do comprometimento com o conteúdo a ser repassado aos alunos. A Tabela 7, contém a definição detalhada de cada atividade deste sub-processo.

Atividade	*Descrição*
Montar a lista de ações	Neste ponto o gerador deverá apresentar as opções de ações já configuradas em outros jogos que existem no banco de dados (de acordo com o nível) para que o professor elabore os detalhes da execução da fase. A lista de ações definirá o tamanho da fase. Uma lista de ações poderá conter ações obrigatórias e também ações opcionais. Uma série de regras de interação entre as ações pode ser estipulada como decisões e efeitos a serem realizados pelo jogo. Cada fase possui uma série de eventos (*AoIniciarNaoExecutada(), AoExecutar(), AoClicarInfoFase(), AoReiniciar(), AoFinalizarSucesso(), AoIniciarExecutada(), AoAtingirPontuacao()*), neste caso cada um desses eventos terá um conjunto de ações configuradas pelo professor em uma sequencia lógica definida por ele.
Criar ação	O professor deverá inicialmente definir os dados básicos da ação como nome, descrição detalhada, os objetivos da ação, em qual(is) tabuleiro(s) se aplica e qual o tipo da ação. O conjunto de ações definidas na fase poderá alterar o comportamento do jogo como um todo, por exemplo uma ação poderá desencadear a liberação de uma outra fase que está definida no cenário principal, porém está bloqueada.
Configurar Componentes das Ações	Cada ação poderá disparar atividades de tipos diferentes. Os tipos são pré-definidos pelo gerador (Exibir Popup, Disponibilizar objeto do baú, Adicionar Recompensa ao Baú, Liberar Fase, Exibir Créditos Fase Finalizada, Trocar Valor Parâmetro Ação, Inicializar fase, Acionar Fase). Uma ação deverá possuir um conjunto de instruções em alto nível definidas pelo professor para realizar o comportamento do jogo utilizando para isto os componentes que estão disponíveis na base de dados.
Criar Componente	Um componente é um objeto estático, cuja seu uso está definido dentro de ações. Um componente neste caso será um artefato que poderá ser utilizado na execução das ações. Dependendo do tipo da ação ela necessitará de um componente como parâmetro. Cada componente criado estará alimentando uma base de dados já existente com modelos de componente. Esta base de dados serve para que próximos cenários sejam montados utilizando componentes de outros.
Definir Ações	O professor deverá definir a ordem de execução das ações para o evento e a fase em questão. A ordenação pode ser configurada para ser: sequencial, aleatória e ainda decisões podem modificar o comportamento da ordenação.
Desenhar Fase	O professor utilizará ferramenta de desenho com um conjunto de objetos padrão para representar as ações e componentes do jogo.

Tabela 7 – Descrição das atividades do sub-processo Definir as Fases (Acervo do Autor).

A seguir, após a definição completa do jogo, o professor deverá definir os participantes. Dependendo do jogo, os participantes podem ser adicionados pelo professor de maneira arbitrária, liberando assim o jogo somente para aqueles que estiverem vinculados. Esta fase é importante principalmente para poder comunicar os alunos sobre o jogo a ser executado. Além disso, o professor poderá configurar equipes para a condução das atividades, sendo que todas essas decisões devem ser tomadas em virtude do tipo de conteúdo comtemplado e o “enredo” do mesmo. Na Figura 21, é apresentada a fase do processo que trata dos participantes.
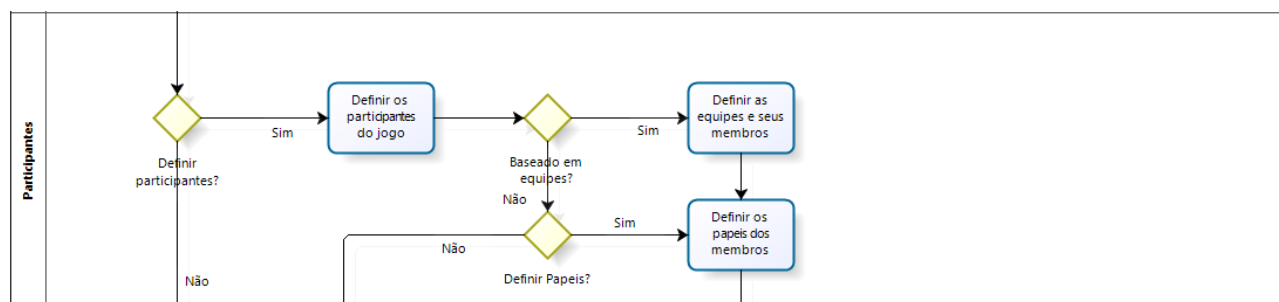
Figura 21 – Atividades da fase de Participantes (Acervo do Autor).

A fase de participantes é opcional, o professor poderá ou não pré-definir quem poderá executar o jogo. Caso o jogo tenha participante pré-definido então ele será um “jogo fechado”, caso contrário será um “jogo aberto”. Na Tabela 9, é apresentada a definição detalhada de cada atividade.

Atividade	*Descrição*
Definir os participantes do jogo	Neste ponto o professor poderá selecionar todos os alunos ou então parte da turma. Os participantes serão envolvidos durante todo o jogo (do inicio ao fim). Cada participante deverá estar previamente cadastrado, incluindo seus dados como Telefone de contato, e-mail, facebook, etc.
Definir as equipes e seus membros	Caso seja um jogo baseado em equipes, então deve-se definir as equipes para o cenário atual. As equipes serão formadas para disputar todas as fases do jogo.
Definir os papéis dos membros	Caso seja optado por definir os papeis previamente então deve-se fazê-lo, com base na estratégia do jogo. Os papeis dependem do contexto do jogo. Se a ideia é fazer apenas um jogo para levantamento e análise, então pode-se definir: Cliente e Analista de Sistema. Os papéis são cadastrados pelo professor de maneira genérica para todos os jogos.

Tabela 9 – Descrição das atividades da fase de Participantes (Acervo do Autor).

Após definidos os participantes, o jogo poderá ser liberado pelo professor para ser executado. A liberação poderá ser do jogo inteiro ou então de cada fase. O professor poderá nesta fase, fazer o acompanhamento do jogo, monitorando as confirmações dos participantes (para poder liberar o jogo ou as fases), monitorando as atividades em execução e realizando intervenções no jogo (como enviar mensagem para o jogador, que ficará disponível na sua caixa de entrada). A Figura 22 demonstra a linha de execução dessa fase.
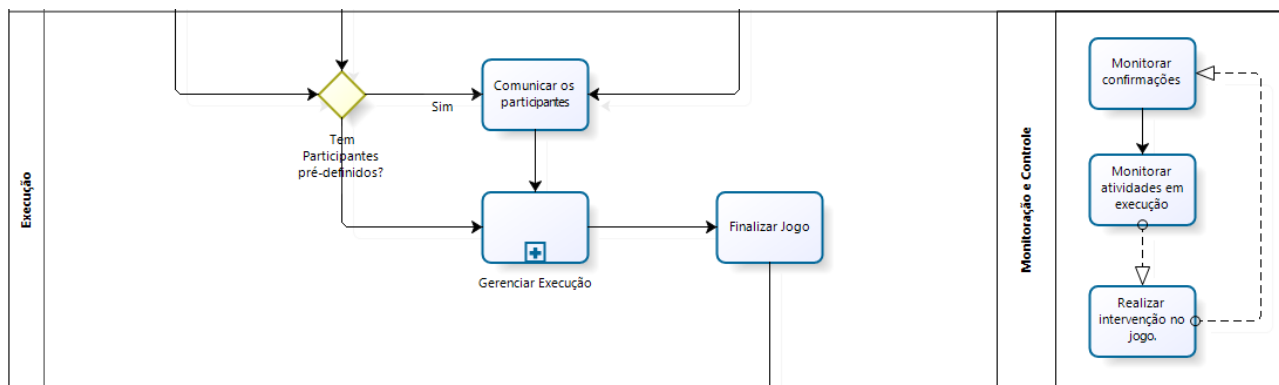
Figura 22 – Atividades da fase de Execução (Acervo do Autor).

Uma observação importante a respeito do fluxo da Figura 22, é referente a subfase de Monitoração e Controle. Esta subfase é executada enquanto o jogo ainda não finalizou e ela não possui um disparo pré-definido, ou seja, o professor poderá realizar as atividades que a compõe a qualquer momento. Na Tabela 10 estão dispostas as atividades e suas definições.

Atividade	*Descrição*
Comunicar os Participantes	Ao iniciar a execução do jogo deve-se inicialmente comunicar os participantes pré-definidos. Será enviado um e-mail para cada participante que deverá imediatamente clicar em um link no e-mail para confirmar. Quando todos os membros confirmarem, o jogo poderá iniciar. Esta atividade é opcional dependendo do uso ou não de participantes pré-definidos.
Gerenciar a Execução	Assim que todos os membros confirmarem, as fases do jogo podem ser iniciadas. Mais detalhes na Tabela 8, descrição das atividades de gerenciamento da execução.
Finalizar Jogo	A finalização do jogo poderá ser executada pelo professor a qualquer momento, não apenas quando todas as fases foram executadas pelos alunos. A finalização do jogo enviará uma comunicação para todos os participantes informando sobre isto.

Tabela 10 – Descrição das atividades da fase de Execução (Acervo do Autor).

Na Tabela 11 estão definidas as atividades da subfase de Monitoração e Controle, a ser realizada pelo professor.

Atividade	*Descrição*
Monitorar Confirmações	O professor deverá num primeiro momento monitorar as confirmações dos alunos. Isto é especialmente importante caso o jogo seja liberado apenas quando todos confirmarem. Poderá a qualquer momento descredenciar um aluno/participante.
Monitorar Atividades em Execução	O gerador será responsável por armazenar um "log" das atividades executadas pelos participantes nos jogos. O professor poderá a qualquer momento verificar as atividades através da monitoração
Realizar intervenção no jogo.	Se necessário, o professor poderá executar ações de controle como enviar mensagem, descredenciar participante, etc.

Tabela 11 – Descrição das atividades da subfase de Monitoração e Controle (Acervo do Autor).

Durante a fase de execução, o sub-processo de gerenciamento da execução (Gerenciar Execução) será executado. Ele somente será executado quando todos os participantes confirmarem o seu vínculo através da atividade "Comunicar Participantes", definida na Tabela 11.

A liberação do jogo e das fases será executada neste sub-processo, conforme pode ser observado na Figura 23.
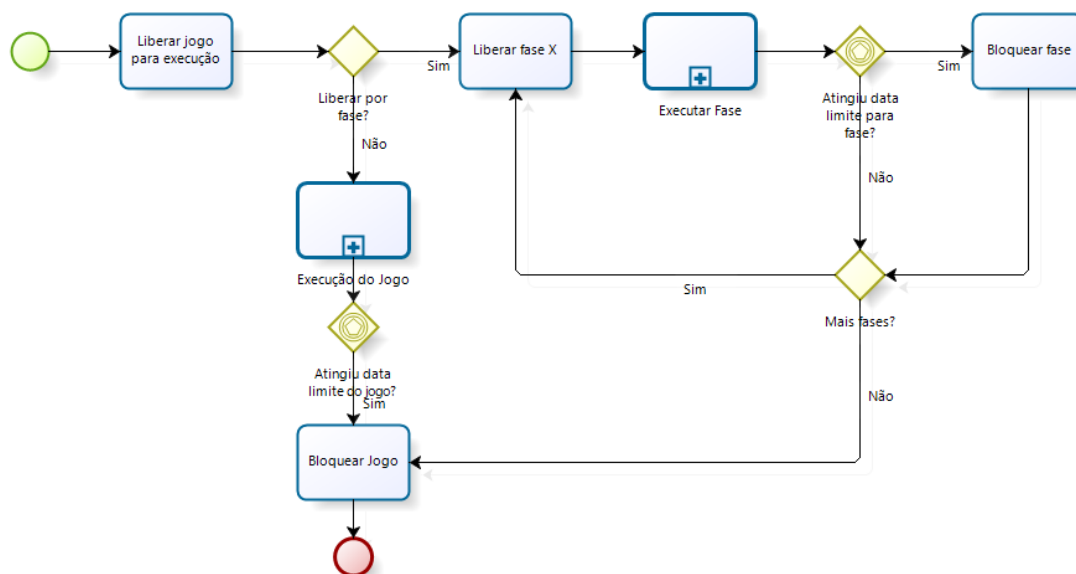
Figura 23 – Atividades do sub-processo Gerenciar Execução (Acervo do Autor).

Importante destacar que os comportamentos de liberar jogo, liberar fase ou bloquear fase e bloquear jogo são realizados pelo professor. Na Tabela 23 estão definidas as atividades do sub-processo Gerenciar Execução, detalhadamente.

A Tabela 12 descreve as atividades do sub-processo de gerenciamento da execução dos jogos.

Atividade	*Descrição*
Liberar Jogo para execução	O professor deverá inicialmente liberar o jogo para execução, desta forma os participantes já poderão iniciar o jogo.
Liberar Fase X	Em seguida da liberação do jogo, o professor poderá optar pela liberação fase a fase.
Executar a fase	Quando liberado por fase, o jogador poderá executar apenas uma a uma. Isto poderá ser útil caso o professor queira que todos os alunos realizem atividades em comum, podendo o professor passar conteúdo sobre o determinado assunto e conforme avança libera novas fases. A execução da fase é realizada pelo aluno/jogador.
Bloquear fase	O professor poderá a qualquer momento realizar o bloqueio de uma determinada fase, sendo que os participantes que não executaram ou não concluíram a fase, não poderão mais acessá-la. O bloqueio da fase também poderá ser automático, quando definida uma data limite.
Execução do Jogo	Quando o professor decide liberar para que o jogo inteiro possa ser jogado, o aluno poderá iniciar e executar todas fazes de uma só vez, executando o sub-processo de execução de fases novamente.
Bloquear o Jogo	O professor poderá bloquear o jogo a qualquer momento. Desta forma o jogo ficará inacessível para os seus participantes. O bloqueio do jogo também poderá ser automático, quando definida uma data limite.

Tabela 12 – Descrição das atividades do Sub-Processo Gerenciar Execução (Acervo do Autor).

O sub-processo de execução do jogo é realizado inteiramente pelo aluno. O mesmo deverá inicialmente proceder com o login no portal principal do gerador, que exibirá uma lista com os jogos disponíveis para ele (jogos abertos ou então que ele esteja como participante), em seguida deverá executar as fases do jogo e posteriormente (assim que todas as fases forem finalizadas pelo aluno), será executado o evento *AoFinalizarCenario*(), que terá seu comportamento definido na configuração do cenário principal.

O sub-processo de execução da fase é executado fase a fase quando o professor optar por liberação por fase, ou então nas formas sequencialmente (exibindo a nova fase conforme o aluno “passa” pela fase anterior) e manualmente (normalmente dispondo as fases em um mapa no cenário principal). Independentemente da forma como é acessada a fase, o seu sub-processo de execução é o mesmo e isto pode ser visualizado na Figura 24.
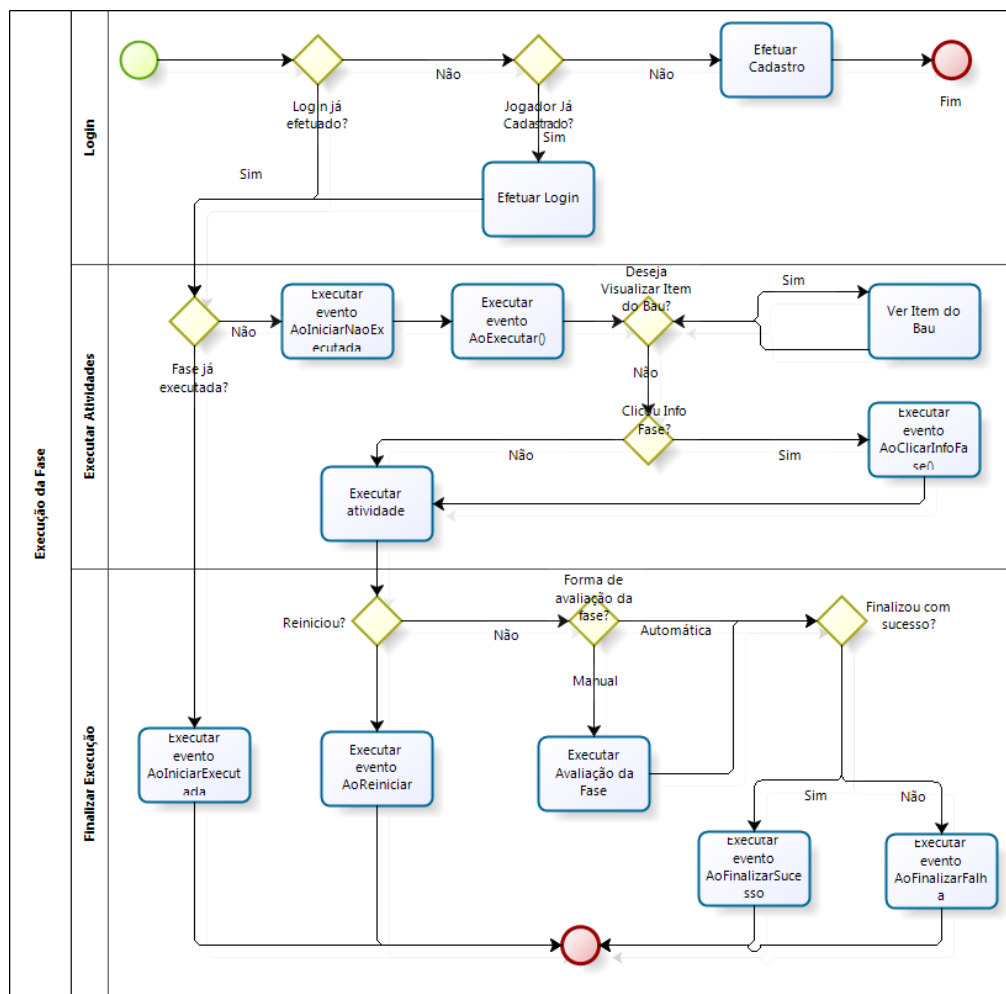
Figura 24 – Atividades do sub-processo Executar Fase (Acervo do Autor).

Importante ressaltar que o processo apresentado na Figura 24, trata da execução de uma única fase por vez. Por este motivo, o sub-processo possui uma fase de Login e uma decisão de “Login Já Efetuado” que poderá conduzi-lo diretamente para a fase de Executar Atividades.

Na Tabela 13 pode-se observar as atividades com a respectiva definição de cada uma delas.

Atividade	*Descrição*
Efetuar Cadastro	O Jogador deverá efetuar o seu cadastro no banco de dados do jogo. O cadastro do jogador poderá ser utilizado em todos os jogos criados e disponibilizados naquela base de dados. Informar seu nome e e-mail, fazer o upload de uma foto sua, informar uma senha e validar a conta acessando URL enviada por e-mail para validar o seu cadastro.
Efetuar Login	Após devidamente cadastrado e com sua conta validada, o aluno poderá então fazer login e visualizar os jogos disponíveis para ele.
Executar evento *AoIniciarNaoExecutada*	O evento é disparado automaticamente, e poderá executar ações para inicialização da fase. Este evento é disparado apenas quando a fase ainda não foi finalizada.
Executar evento *AoExecutar*	O evento é disparado automaticamente sempre que a fase é iniciada.
Ver Item do Baú	Caso itens do baú estejam disponíveis ao aluno, o mesmo poderá visualizar. Neste caso os itens do baú serão disponibilizados através da execução de ações do tipo "Disponibilizar Objeto do Baú", normalmente no evento *AoIniciarNaoExecutada*() e conforme disponibilidade desses itens no baú. Exemplos de itens são as dicas, dados da sorte, itens de recompensa, etc. Os objetos ficarão visíveis em uma área de fácil visualização do jogador em cada fase.
Executar Evento *AoClicarInfoFase*	O evento é disparado quando o jogador decidir acionar a opção de informações da fase, que são definidas pelo professor ao construir a fase. O evento poderá ser disparado quantas vezes o aluno desejar.
Executar Atividade	O jogador deverá executar a atividade conforme solicitado na interface da fase do jogo. As atividades levarão em conta o tipo de cada fase definida.
Executar Evento *AoReiniciar*	O evento *AoReiniciar* será disparado automaticamente, quando o jogador acionar a opção “Reiniciar” (quando disponível através de configuração da fase) e poderá acionar diversas sub ações pré-definidas pelo professor.
Executar Avaliação da Fase	A avaliação da fase é a ação que o aluno dispara no gerador para que a avaliação da fase seja executada para verificar se a fase foi finalizada com sucesso ou não. Isto dependerá do tipo de fase: Fases de seleção de opções de múltipla escolha devem ser acionadas manualmente.
Executar Evento *AoFinalizarSucesso*	O evento pode ser disparado automaticamente quando o aluno conseguiu executar a fase com sucesso, dependendo do tipo da fase, por exemplo, fases de respostas de múltipla escolha o aluno deverá acionar uma opção para avaliação da fase. A execução da fase com sucesso ou não dependerá do tipo de fase escolhida pelo professor.
Executar Evento *AoFinalizarFalha*	Este evento é disparado quando o número de tentativas restante para a fase for alcançado.

Tabela 13 – Descrição das atividades do Sub-Processo Executar Fase (Acervo do Autor).

APÊNDICE B – GDD - A Ilha dos Requisitos

Gerador de Jogo Sério

Game Design Document

A Ilha dos Requisitos

08/01/2014

<<Nome do Professor>>

Revisões

Versão	Professor	Descrição da Versão	Data da Versão
1.0	<< >>	Versão inicial do jogo A Ilha dos Requisitos	08/01/2014

Sumário

A História do Jogo

Enquanto isto em um voo da CrashTour sobre o atlântico.... Jack Reqs, um analista de requisitos iniciante, estava em sua primeira viagem a trabalho... Quando algo dá errado e o avião começa a cair. Os pilotos conseguem escapar, mas.... Jack Reqs estava lendo seus e-mails e acabou caindo com o avião em uma ilha desconhecida.

Definições do Jogo

Dados Básicos

- **Nome do Jogo**: A Ilha dos Requisitos
- **Objetivos**:

Objetivo	Descrição
1	Conhecer as fases do processo de engenharia de requisitos
2	Conhecer as habilidades de um analista de requisitos
3	Praticar elicitação de Requisitos e entender quais são os reais problemas
4	Classificação de requisitos
5	Validação de Requisitos
6	Conhecer as atividades do gerenciamento de mudanças

- **Data Limite para execução**: 31/12/2014.

Nível de Dificuldade

- **Nível do Jogo**: Iniciante
- **Pré-requisitos**:
  a) conhecimentos gerais sobre engenharia de software;
  b) conhecimentos mínimos sobre engenharia de requisitos;

Definições do Modo de Jogo

- **Modo de Jogo**: Fases posicionadas no Mapa
- **Participantes pré-definidos**: Não. (Jogo Aberto).
- **Jogo em equipes**: Não.
- **Contra o Tempo**: Sim.
- **Permitir Envio de Mensagens entre os Participantes**: Não

Baú de Recompensas

- **Pontuação Mínima**: Não há.
- **Pontuação Inicial**: 0
- **Artefato de Valor:** "Dicas Restantes" no plural e "Dica restante" no singular, que poderão ser utilizadas pelo jogador em diferentes fases, sendo que a localização de cada dica será definida pelo professor. Tipo dica, saldo inicial 6. Pontuação: 150 pontos (cada uma)

Listagem de Dicas	
ID Dica	Descrição
1	Na fase de Análise devem ser resolvidos os conflitos provenientes da fase de elicitação.
2	Após a verificação do documento de requisitos gerado na fase de especificação, o cliente deve validar os requisitos.
3	Controlar o custo e o tempo do projeto são responsabilidades do gerente de projeto.
4	O Analista de Requisitos deve mediar discussões e conflitos entre os stakeholders
5	O problema deve ser expresso de maneira desacoplada da solução.
6	Um dos problemas de Jack é ter que salvar o restante da tribo
7	Os requisitos não funcionais referem-se as restrições que a solução deve atender.
8	Os requisitos funcionais referem-se as funcionalidades que a solução deve apresentar
9	Durante a validação podem ser identificados novos requisitos.
10	Antes de determinar se uma mudança é viável é necessário analisar o seu impacto.
11	Para sair da ilha com os Nerds e os Geeks pode ser necessário incluir ou alterar os requisitos já existentes

- **Artefato de Valor:** "Dias restantes" no plural e "Dia restante" no singular. Tipo inteiro, saldo inicial de 22. Pontuação: 0. Evento: aoZerar(FIM_DE_JOGO).
- **Artefato de Valor:** "Tempo restante" (plural e singular). Tipo: Minuto Restante, saldo inicial 45. Pontuação: 60 pontos cada minuto. Evento: aoZerar(FIM_DE_JOGO).
- **Recompensa**: "Bateria do Notebook". Deverá existir uma recompensa na fase 10 (Avião) onde ao acessá-la o jogador ganhará a bateria do notebook de Jack. Saldo Inicial 0. Pontuação: 100 pontos
- **Artefato de Valor**: "Dado da Sorte" (plural e singular). Tipo: Dado da Sorte, saldo inicial: 1.
* O dado da sorte tem seu comportamento realizado dependendo do tipo de fase. Se for lista de opções, irá marcar as opções corretas na ordem sequencial dependendo do número sorteado. Se for arranjo de objetos, irá organizá-los da mesma forma. Se for classificação também.

Regras do Jogo

ID Regra	Descrição
1	Você tem 22 dias (45 minutos no tempo real) para terminar o jogo.
2	Durante o jogo, você deverá enfrentar diversos desafios. Para cada tentativa de resolução de um desafio, 1 dia será consumido
3	Você deverá utilizar o notebook de Jack (protagonista do jogo) para consultar dicas. Porém antes, você deverá encontrar uma bateria para utilizá-lo
4	Os desafios estão distribuídos pelo mapa. Logo, procure explorar o cenário com cuidado.
5	Para entrar em uma área específica do mapa utilize um clique duplo.
6	Utilize o tempo que for necessário para ler as instruções. Durante o tempo de leitura, o tempo do jogo ficará "congelado"

Definições Básicas do Cenário Principal

- **Descrever o cenário**: Jack Reqs caiu em uma ilha de canibais chamada "A Ilha dos requisitos". Jack é um analista de sistemas e utilizando apenas o seu conhecimento na área ele deverá escapar da ilha.
- **Imagem de Fundo**: A imagem de fundo será a mesma do jogo original estática.
- **Fases serão iniciadas automaticamente**: Não (desta forma o jogador deverá clicar sobre as fases no cenário).
- **Disposição visual das fases no cenário**: Automática (posteriormente deverá utilizar a ferramenta de desenho para posicionar as fases no cenário).

Definição das Fases

- **Fases**: Ao todo serão 10 as fases do jogo.
- **Pontuação por Fase**: Cada fase vencida valerá 100 pontos.
- **Reinicio**: Cada reinicio de fase vale: -20 pontos

1. **Nome: NERDS**. (Organizar Arranjo de Objetos na Sequencia Correta) Deverá identificar quais são as fases da engenharia de requisitos através da correta colocação das pedras nas caixas de acordo com a ordem desejada. Obrigatória.
2. **Nome: Pedra Sagrada 1**. (Selecionar as Opções Corretas, múltipla escolha) Deverá identificar quais são as habilidades associadas com o analista de requisitos. Obrigatória.
3. **Nome**: **Floresta**. (Selecionar as Opções Corretas, múltipla escolha) Deverá identificar quais são os seus reais problemas. Obrigatória
4. **Nome**: Repita a sequencia (Fase Bônus)
   Esta fase bônus disponibilizará ao jogador a opção de repetir a sequencia de cores para ganhar mais pontos no seu baú de recompensa.
5. **Nome**: **Pedra Sagrada 2**. (Classificar objetos semelhantes) Deverá classificar os requisitos funcionais de não funcionais. Obrigatória.
6. **Nome**: **Praia 1.** (Selecionar as Opções Corretas, quadro visual) O jogador deverá responder quais são as atividades necessárias para validação dos requisitos para a construção do barco. Deverá também informar. Obrigatória.
7. **Nome**: **Geeks.** (Organizar Arranjo de Objetos na Sequencia Correta) Deverá identificar quais é o ciclo de vida de uma solicitação de mudança de requisitos. Obrigatória.
8. **Nome:** Mercado (Mercado de Coisas)

Esta fase bônus, estará disponível caso o jogador consiga atingir 600 pontos até o momento. O jogador poderá comprar uma coisa no mercado caso ele tenha encontrado a bateria do notebook de Jack no jogo.

9. **Nome: Praia 2** (Lista de Opções, múltipla escolha). Informar o que é necessário fazer para sair da ilha com todos os Nerds e Geeks. Ações após a solicitação de mudança. Obrigatória.
10. **Nome: Avião** (Apenas visitar). Ao visitar esta fase será adicionado ao baú de recompensas uma recompensa "Bateria do Notebook"

Participantes do Jogo

- **Participantes**: Como o jogo será aberto, não será definida uma lista de participantes.

Nome	E-Mail

Liberação do Jogo

- **Método de Liberação**: A partir desse momento o jogo poderá ser executado por quem quiser.

Bloqueio do Jogo

- **Método de Bloqueio**: Ao atingir a data limite, deverá bloquear a execução do jogo.

Finalização do Jogo

- **Método de Finalização do Jogo:** O Professor irá finalizar o jogo manualmente ou quando o mesmo atingir a data limite.

Definição do cenário principal

Configuração das Ações

Lista de Ações do evento: aoIniciarCenario()	
Seq. Ação	Definição
1	**Tipo:** Carregar Informações da Seção de Jogo * Serve para carregar os dados de jogo anterior do usuário.
2	**Tipo:** Exibir Popup Imagem **Imagem**: img_quadrinhos

Lista de Ações do evento: aoFinalizarCenario()	
Seq. Ação	Definição
1	**Tipo:** Avaliar Item do Baú **Tipo do Item**: Recompensa **Item**: "Bateria do Notebook" **Necessário**: Sim
2	**Tipo:** Exibir Imagem **Imagem**: img_credito_fim_jogo

Lista de Componentes do Cenário

Componente	**Definição**
img_tribo_nerds	**Tipo**: Objeto Imagem (img_tribo_nerds.jpg) **Evento**: aoClicar() – exibir_info_fase(1) **Evento**: aoClicarDuplo() – acessar_fase(1)
img_pedra_sagrada	**Tipo:** Objeto Imagem (img_pedra_sagrada.jpg) **Evento**: aoClicar() – exibir_info_fase(2) **Evento**: aoClicarDuplo() – acessar_fase(2)
img_floresta	**Tipo:** Objeto Imagem (img_floresta.jpg) **Evento**: aoClicar() – exibir_info_fase(3) **Evento**: aoClicarDuplo() – acessar_fase(3)
img_praia	**Tipo**: Objeto Imagem (img_praia.jpg) **Evento**: aoClicar() – exibir_info_fase(6) **Evento**: aoClicarDuplo() – acessar_fase(6)
img_tribo_geek	**Tipo**: Objeto Imagem (img_tribo_geek.jpg)

	Evento: aoClicar() – exibir_info_fase(7) **Evento**: aoClicarDuplo() – acessar_fase(7)
img_aviao	**Tipo**: Objeto Imagem (img_aviao.jpg) **Evento**: aoClicar() – exibir_info_fase(10) **Evento**: aoClicarDuplo() – acessar_fase(10)
img_quadrinhos	**Tipo**: Objeto Imagem (quadrinhos.jpg)
img_mercado	**Tipo:** Objeto Imagem (img_mercado.jpg)
img_bonus	**Tipo:** Objeto Imagem (img_bonus.jpg)
jack_raqs	**Tipo:** Personagem do Jogo **Imagem**: img_jack.jpg

** O Posicionamento dos componentes no cenário principal, será feito posteriormente através da ferramenta de desenho.*

Definição das fases

<table>
<tr><td>Fase</td><td>Definição</td></tr>
<tr><td>1</td><td>
Imagem de Fundo: img_area_nerds
Obrigatória: Sim.
Bloqueada: Não.
Pode voltar ao cenário principal: Sim.
Permite Reiniciar: Sim
Número de Tentativas Máximo: 6
Info Fase: Área NERDS
Tipo Fase: Organizar Arranjo de Objetos na Sequencia Correta
Créditos Fase Finalizada: “Agora, Jack é venerado como Byte, o Salvador, pelos membros da Tribo Nerd...” + “Parabéns!! Você conseguiu salvar a pele de Jack!!”
<table>
<tr><td colspan="2">Lista de Ações do evento: AoIniciarNaoExecutada()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Exibir Popup
Descrição: “Jack estava desbravando a ilha, quando acabou capturado pela tribo canibal dos Nerds...”
Parar Tempo Restante: Sim
Evento: aoClicar() – fechar_poppup</td></tr>
<tr><td>2</td><td>Tipo: Exibir Popup
Descrição: “Este desafio é sobre o processo de engenharia de requisitos. Não sabemos como, mas a tribo dos Nerds parece estar muito relacionada com esta área da computação...”
Parar Tempo Restante: Sim
Evento: aoClicar() – fechar_popup</td></tr>
<tr><td>3</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dica(1)
Ativo: Sim</td></tr>
<tr><td>4</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dica(2)
Ativo: Sim</td></tr>
<tr><td>5</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Bateria do Notebook
Ativo: Se não existir</td></tr>
<tr><td>6</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dias restantes
Ativo: Sim</td></tr>
<tr><td>7</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dado da Sorte
Ativo: Se existir</td></tr>
</table>
<table>
<tr><td colspan="2">Lista de Ações do evento: AoExecutar()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Inicializar fase.
- Exibir imagem de fundo</td></tr>
</table>
</td></tr>
</table>

Lista de Ações do evento: AoClicarInfoFase()	
Seq. Ação	Definição
1	**Tipo**: Exibir Popup **Descrição**: “Clique em uma das pedras, arraste e solte no espaço....” **Parar Tempo Restante**: Sim **Evento**: aoClicar() – fechar_popup

Lista de Ações do evento: AoReiniciar()	
Seq. Ação	Definição
1	**Tipo:** Remover Item do Baú **Objeto:** Dias restantes ➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase executada
2	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup
3	**Tipo:** Liberar Fase **Fase: 2.**
4	**Tipo:** Trocar Valor Parâmetro Ação **Objeto:** img_praia **Ação:** acessar_fase **Novo Valor**: 6

Lista de Ações do evento: AoIniciarExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

Lista de Componentes da Fase	
Nome	**Definição**
img_elicitacao	**Tipo**: Objeto Imagem **Imagem**: img_elicitacao.jpg
img_analise	**Tipo:** Objeto Imagem **Imagem**: img_analise.jpg
img_especificacao	**Tipo:** Objeto Imagem **Imagem:** img_especificacao.jpg
img_verificacao	**Tipo:** Objeto Imagem **Imagem:** img_verificacao.jpg
img_validacao	**Tipo:** Objeto Imagem **Imagem:** img_validacao.jpg
img_gerencia	**Tipo:** Objeto Imagem **Imagem:** img_gerencia.jpg

Lista de Opções de Arranjo	
Nome	**Definição**
Elicitação	**Posição Inicial**: Aleatória **Posição Ordenada**: 1 **Imagem Objeto**: img_elicitacao
Análise	**Posição Inicial**: Aleatória **Posição Ordenada**: 2 **Imagem Objeto**: img_analise
Especificação	**Posição Inicial**: Aleatória **Posição Ordenada**: 3

	Imagem Objeto: img_especificacao
Verificação	**Posição Inicial**: Aleatória **Posição Ordenada**: 4 **Imagem Objeto**: img_verificacao
Validação	**Posição Inicial**: Aleatória **Posição Ordenada**: 5 **Imagem Objeto**: img_validacao
Gerência	**Posição Inicial**: Aleatória **Posição Ordenada**: 6 **Imagem Objeto**: img_gerencia

2

Imagem de Fundo: img_pedra_sagrada
Obrigatória: Sim.
Bloqueada: Sim.
Pode voltar ao cenário principal: Sim.
Permite Reiniciar: Sim
Número de Tentativas Máximo: 6
Info Fase: Área Pedra Sagrada
Tipo Fase: Selecionar as Opções Corretas
Lista Aleatória: Sim
Forma: Lista de Múltipla Escolha
Descrição do Problema: “Marque apenas as habilidades associadas com um analista de requisitos”
Créditos Fase Finalizada: “Paragens!!...”

Ordem	**Opção**	**Correta?**
1	Definir atividades e prazos (cronograma) de um projeto	Não
2	Ser um bom ouvinte	Sim
3	Elaborar os casos de teste	Não
4	Monitorar e controlar o esforço gasto com requisitos	Não
5	Analisar o problema do usuário e identificar suas necessidades	Sim
6	Executar os testes funcionais	Não
7	Mediar discussões e conflitos	Sim
8	Gerenciar custos de um projeto	Não
9	Ser um bom observador	Sim

Lista de Ações do evento: AoIniciarNaoExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Popup **Descrição:** “Agora a pedra sagrada confia nas habilidades de Jack Reqs, e irá ajuda-lo a transmitir os requisitos necessários para sair da ilha para os membros da tribo dos NERDS” **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_poppup
2	**Tipo:** Exibir Popup **Descrição:** “Este desafio é sobre o processo de engenharia de requisitos. Não sabemos como, mas a tribo dos Nerds parece estar muito relacionada com esta área da computação...” **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_popup
3	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(3) **Ativo:** Sim
4	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(4) **Ativo:** Sim
5	**Tipo:** Disponibilizar objeto do baú **Objeto:** Bateria do Notebook **Ativo:** Se não existir

6	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dias restantes **Ativo:** Sim
7	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dado da Sorte **Ativo:** Se existir

Lista de Ações do evento: AoExecutar()	
Seq. Ação	Definição
1	**Tipo**: Inicializar fase. - Exibir imagem de fundo

Lista de Ações do evento: AoReiniciar()	
Seq. Ação	Definição
1	**Tipo:** Remover Item do Baú **Objeto:** Dias restantes ➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase executada
2	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup
3	**Tipo:** Liberar Fase **Fase: 3.**
4	**Tipo:** Trocar Valor Parâmetro Ação **Objeto:** img_pedra_sagrada **Ação:** acessar_fase **Novo Valor**: 4

Lista de Ações do evento: AoIniciarExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

3

Imagem de Fundo: img_floresta
Obrigatória: Sim.
Bloqueada: Sim.
Pode voltar ao cenário principal: Sim.
Permite Reiniciar: Sim
Número de Tentativas Máximo: 6
Info Fase: Área Floresta
Tipo Fase: Selecionar as Opções Corretas
Forma: Lista de Múltipla Escolha
Lista Aleatória: Sim
Descrição do Problema: "Marque as frases que identificam o problema de Jack"
Créditos Fase Finalizada: "Parabéns!! Você conseguiu ajudar Jack a reconhecer qual o real problema a ser resolvido!! Para entender melhor a sua resposta veja os comentários abaixo:..."

Ordem	**Opção**	**Correta?**
1	O vulcão irá explodir a ilha	Sim
2	Precisamos sair da ilha	Não
3	Poderíamos nos esconder em uma caverna profunda e aguardar a erupção passar	Não
4	A erupção acontecerá em menos de 30 dias	Sim
5	O avião deverá ser consertado	Não
6	Será necessário armazenar comida para vários dias	Não

7	Vamos construir um barco com capacidade para 30 pessoas	Não
8	Precisaremos de remos longos	Não
9	Para aplacar a sede do Deus bit, podemos fazer um sacrifício	Não
10	Uma grande vela poderia ajudar a quebrar a rebentação	Não
11	A tribo dos Nerds tem 15 membros, incluindo homens, mulheres e crianças	Sim

Lista de Ações do evento: AoIniciarNaoExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Popup **Descrição:** "O grande vulcão BigBug está para entrar em erupção!!! Segundo a lenda, a explosão final será 30 dias após a chegada de Byte, o Salvador!! O tempo está correndo..." **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_poppup
2	**Tipo:** Exibir Popup **Descrição:** "Neste desafio Jack listou uma série de frases para tentar identificar qual é o problema a ser resolvido. Ajude-o a identificar quais frases correspondem ao problema" **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_popup
3	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(5) **Ativo:** Sim
4	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(6) **Ativo:** Sim
5	**Tipo:** Disponibilizar objeto do baú **Objeto:** Bateria do Notebook **Ativo:** Se não existir
6	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dias restantes **Ativo:** Sim
7	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dado da Sorte **Ativo:** Se existir

Lista de Ações do evento: AoReiniciar()	
Seq. Ação	Definição
1	**Tipo:** Remover Item do Baú **Objeto:** Dias restantes ➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase executada
2	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup
3	**Tipo:** Liberar Fase **Fase: 4.**
4	**Tipo:** Liberar Fase **Fase: 5.**

Lista de Ações do evento: AoIniciarExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

4	**Imagem de Fundo**: img_bonus **Obrigatória**: Não. **Bloqueada**: Sim. **Pode voltar ao cenário principal**: Sim. **Permite Reiniciar**: Sim **Número de Tentativas Máximo**: 6 **Info Fase**: Área Bônus **Tipo Fase**: Bônus **Forma**: Repita a sequencia **Forma Imagem**: Quadrado **Créditos Fase Finalizada**: “Você já conseguiu resgatar o bônus desta fase” **Descrição do Problema**: “Segundo o RUP o ciclo de vida de um sistema consiste em quatro fases:”

Lista de Ações do evento: AoIniciarNaoExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Popup **Descrição:** “Esta fase bônus poderá render mais conhecimento ao Jack, isto resultará em mais 50 pontos para o seu baú de recompensas. Este bônus consiste em repetir a sequencia até que o último item seja exibido” **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_poppup

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase Bônus **Valor**: 100 pontos
2	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup

Lista de Ações do evento: AoIniciarExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

Lista de Itens da Sequencia	
Sequencia	**Definição**
1	**Nome**: Concepção **Cor Relacionada**: Verde
2	**Nome:** Elaboração **Cor Relacionada:** Azul
3	**Nome:** Construção **Cor Relacionada:** Vermelho
4	**Nome:** Transição **Cor Relacionada**: Preto

5	**Imagem de Fundo**: img_pedra_sagrada **Obrigatória**: Sim. **Bloqueada**: Sim. **Pode voltar ao cenário principal**: Sim. **Permite Reiniciar**: Sim **Número de Tentativas Máximo**: 6 **Info Fase**: Área Pedra Sagrada **Tipo Fase**: Classificar objetos semelhantes **Créditos Fase Finalizada**: “Agora que os requisitos estão classificados, você deverá ajudar Jack a decidir o que fazer em seguida...”

Lista de Ações do evento: AoIniciarNaoExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Popup

	Descrição: “Depois de ter sido reconhecido pela Pedra Sagrada como Byte o Salvador, Jack já pode passar os requisitos para a tribo dos Nerds. Jack iniciou uma jornada pela Ilha dos requisitos para avaliar os recursos disponíveis e identificar um conjunto de requisitos para resolver o problema...” **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_poppup
2	**Tipo:** Exibir Popup **Descrição:** “Neste desafio Jack espalhou os requisitos que ele já identificou no chão da floresta. Ele também reservou um espaço para guardar os requisitos funcionais e os não funcionais. Você deverá clicar e arrastar cada requisito para o local adequado” **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_popup
3	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(7) **Ativo:** Sim
4	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(8) **Ativo:** Sim
5	**Tipo:** Disponibilizar objeto do baú **Objeto:** Bateria do Notebook **Ativo:** Se não existir
6	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dias restantes **Ativo:** Sim
7	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dado da Sorte **Ativo:** Se existir

Lista de Ações do evento: AoExecutar()	
Seq. Ação	Definição
1	**Tipo**: Inicializar fase. - Exibir imagem de fundo

Lista de Ações do evento: AoClicarInfoFase()	
Seq. Ação	Definição
1	**Tipo**: Exibir Popup **Descrição**: “Para classificar um requisito você deverá arrastá-lo e soltá-lo no local indicado de acordo com sua classificação.” **Parar Tempo Restante**: Sim **Evento**: aoClicar() – fechar_popup

Lista de Ações do evento: AoReiniciar()	
Seq. Ação	Definição
1	**Tipo:** Remover Item do Baú **Objeto:** Dias restantes ➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase executada
2	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup
3	**Tipo:** Liberar Fase **Fase: 6.**

Lista de Ações do evento: AoIniciarExecutada()

Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

Lista de Grupos de Classes de Objetos	
Nome	**Definição**
Requisitos funcionais	**ID do Grupo:** 1
Requisitos não funcionais	**ID do Grupo:** 2

Lista de Opções de Arranjo	
Nome	**Definição**
O barco deve ter capacidade para 10 pessoas	**Posição Inicial**: Aleatória **Grupo Destino**: 2
O barco deve ser de madeira	**Posição Inicial**: Aleatória **Grupo Destino**: 2
O barco deve ter um leme	**Posição Inicial**: Aleatória **Grupo Destino**: 1
O barco deve ter uma vela	**Posição Inicial**: Aleatória **Grupo Destino**: 1
O barco deve estar pronto em menos de 20 dias	**Posição Inicial**: Aleatória **Grupo Destino**: 2
O barco deve ser resistente para durar 5 dias no mar	**Posição Inicial**: Aleatória **Grupo Destino**: 2
O barco deve ter 8 remos	**Posição Inicial:** Aleatória **Grupo Destino**: 1
O barco deve ter uma caixa para armazenar comida	**Posição Inicial:** Aleatória **Grupo Destino:** 1

6

Imagem de Fundo: img_praia
Obrigatória: Sim.
Bloqueada: Sim.
Pode voltar ao cenário principal: Sim.
Permite Reiniciar: Sim
Número de Tentativas Máximo: 6
Info Fase: Área Praia
Tipo Fase: Selecionar as Opções Corretas
Forma: Quadro Visual de Opções`
Lista Aleatória: Sim
Descrição do Problema: "O que você veio fazer na tribo dos NERDS"
Créditos Fase Finalizada: "Paragens!! Você conseguiu ajudar Jack a reconhecer qual o real problema a ser resolvido!! Para entender melhor a sua resposta veja os comentários abaixo:..."

Ordem	**Opção**	**Correta?**
1	Avaliar se existem novos requisitos	Sim
2	Buscar madeira para construção do barco	Não
3	Validar os requisitos documentados na pedra sagrada com UbaapNerd	Sim
4	Descansar um pouco antes da construção do barco	Não
5	Recrutar mais NERDS para a construção do barco	Não
6	Gerenciar a construção do barco	Não

Lista de Ações do evento: AoIniciarNaoExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Popup **Descrição:** "Você esqueceu de lembrar Jack para validar os requisitos com UbaapNerd na tribo dos nerds. Jack acabou perdendo 5 importantes dias. Vá até a tribo dos Nerds agora para conversar com UbaapNerd." **Parar Tempo Restante**: Sim **Evento:** aoClicar() – fechar_poppup
2	**Tipo:** Trocar Imagem Fundo Fase

	Imagem: img_area_nerds
3	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(9) **Ativo:** Sim
4	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(10) **Ativo:** Sim
5	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dias restantes **Ativo:** Sim
6	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dado da Sorte **Ativo:** Se existir

Lista de Ações do evento: AoExecutar()	
Seq. Ação	Definição
1	**Tipo**: Inicializar fase. - Exibir imagem de fundo

Lista de Ações do evento: AoReiniciar()	
Seq. Ação	Definição
1	**Tipo:** Remover Item do Baú **Objeto:** Dias restantes ➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Acionar Fase **Fase:** 6.1

Lista de Ações do evento: AoIniciarExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

6.1

Imagem de Fundo: img_area_nerds
Obrigatória: Sim.
Bloqueada: Sim.
Pode voltar ao cenário principal: Sim.
Permite Reiniciar: Não
Número de Tentativas Máximo: 6
Tipo Fase: Selecionar as Opções Corretas
Forma: Lista de Múltipla Escolha
Fase Pai: 5
Lista Aleatória: Sim
Descrição do Problema: "Você acredita que validar os requisitos com UbaapNerd é importante para?"
Créditos Fase Finalizada: "Parabens!!! Um fator de sucesso para um projeto é que os requisitos realmente atendam as necessidades de seus clientes e usuários. A apresentação e validação dos requisitos com UbaapNerd foi fundamental para identificar um problema nos requisitos levantados por Jack..."

Ordem	**Opção**	**Correta?**
1	Confirmar se o trabalho que será desenvolvido corresponde ao que o cliente deseja?	Sim
2	Informar UbaapNerd sobre os requisitos que serão desenvolvidos	Não
3	Motivar UbaapNerd a ir trabalhar na construção do barco	Não
4	Deixar UbaapNerd atualizado quando ao andamento do projeto	Não
5	Evitar que UbaapNerd peça novos requisitos	Não

<table>
<tr><td></td><td>
<table>
<tr><td colspan="2">Lista de Ações do evento: AoFinalizarSucesso()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Adicionar Recompensa ao Baú
Tipo Recompensa: Fase executada</td></tr>
<tr><td>2</td><td>Tipo: Exibir Créditos Fase Finalizada
Evento: aoClicar() – fechar_popup</td></tr>
<tr><td>3</td><td>Tipo: Liberar Fase
Fase: 7.</td></tr>
<tr><td>4</td><td>Tipo: Trocar Valor Parâmetro Ação
Objeto: img_praia
Ação: acessar_fase
Novo Valor: 9</td></tr>
</table>
</td></tr>
<tr><td>7</td><td>
Imagem de Fundo: img_area_geeks
Obrigatória: Sim.
Bloqueada: Sim.
Pode voltar ao cenário principal: Sim.
Permite Reiniciar: Sim
Número de Tentativas Máximo: 6
Info Fase: Área Geeks
Tipo Fase: Organizar Arranjo de Objetos na Sequencia Correta
Créditos Fase Finalizada: “”
<table>
<tr><td colspan="2">Lista de Ações do evento: AoIniciarNaoExecutada()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Exibir Popup
Descrição: “A tribo dos geeks pediu trégua para UbaapNerd!! Eles desejam também escapar da ilha, mas antes, eles precisam acreditar em Jack. Para isso Jack deverá provar que é Byte o Salvador....”
Evento: aoClicar() – fechar_poppup</td></tr>
<tr><td>2</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dica(11)
Ativo: Sim</td></tr>
<tr><td>3</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Bateria do Notebook
Ativo: Se não existir</td></tr>
<tr><td>4</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dias restantes
Ativo: Sim</td></tr>
<tr><td>5</td><td>Tipo: Disponibilizar objeto do baú
Objeto: Dado da Sorte
Ativo: Se existir</td></tr>
</table>
<table>
<tr><td colspan="2">Lista de Ações do evento: AoExecutar()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Inicializar fase.
- Exibir imagem de fundo</td></tr>
</table>
<table>
<tr><td colspan="2">Lista de Ações do evento: AoClicarInfoFase()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Exibir Popup
Descrição: “Considerando o fluxo de uma mudança de requisitos, coloque as pedras na sequencia correta.”
Evento: aoClicar() – fechar_popup</td></tr>
</table>
<table>
<tr><td colspan="2">Lista de Ações do evento: AoReiniciar()</td></tr>
<tr><td>Seq. Ação</td><td>Definição</td></tr>
<tr><td>1</td><td>Tipo: Remover Item do Baú
Objeto: Dias restantes</td></tr>
</table>
</td></tr>
</table>

	➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase executada
2	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup
3	**Tipo:** Liberar Fase **Quando**: Existir Item no baú(Bateria do Notebook) **Fase: 8.**
4	**Tipo:** Liberar Fase **Fase: 9.**

Lista de Ações do evento: AoIniciarExecutada()	
Seq. Ação	Definição
1	**Tipo:** Exibir Créditos Fase Finalizada **Evento:** aoClicar() – fechar_poppup

Lista de Componentes da Fase	
Nome	**Definição**
img_avaliacao_impacto	**Tipo**: Objeto Imagem **Imagem**: img_avaliacao_impacto.jpg
img_revisao_planejamento	**Tipo**: Objeto Imagem **Imagem**: img_revisao_planejamento.jpg
img_solicitacao_mudanca	**Tipo**: Objeto Imagem **Imagem**: img_solicitacao_mudanca.jpg
img_revisar_produtos	**Tipo**: Objeto Imagem **Imagem**: img_revisar_produtos.jpg
img_revisar_produtos	**Tipo**: Objeto Imagem **Imagem**: img_revisar_produtos.jpg

Lista de Opções de Arranjo	
Nome	**Definição**
Avaliação do Impacto da Mudança	**Posição Inicial**: Aleatória **Posição Ordenada**: 2 **Imagem Objeto**: img_avaliacao_impacto
Revisão do Planejamento (obter comprometimento dos envolvidos)	**Posição Inicial**: Aleatória **Posição Ordenada**: 4 **Imagem Objeto**: img_revisao_planejamento
Solicitação de Mudança	**Posição Inicial**: Aleatória **Posição Ordenada**: 1 **Imagem Objeto**: img_solicitacao_mudanca
Revisar produtos de trabalho relacionados e revisar rastreabilidade	**Posição Inicial**: Aleatória **Posição Ordenada**: 5 **Imagem Objeto**: img_revisar_produtos
Determinação da viabilidade	**Posição Inicial**: Aleatória **Posição Ordenada**: 3 **Imagem Objeto**: img_determinar_viabilidade

8	**Imagem de Fundo**: img_mercado **Obrigatória**: Não. **Bloqueada**: Sim. **Bônus**: Sim **Pode voltar ao cenário principal**: Sim. **Permite Reiniciar**: Não **Número de Tentativas Máximo**: 1 **Info Fase**: Área Mercado **Tipo Fase**: Mercado

	Forma: Quadro Visual de Opções **Descrição do Problema**: "Jack poderá utilizar uma ferramenta para gestão de requisitos, o que fará com que seu trabalho seja mais eficiente e ágil. Compre no mercado a ferramenta desejada para Jack" **Créditos Fase Finalizada**: "Você já comprou a ferramenta para Jack!"

Ordem	**Opção**	**Valor Pontos**	**Beneficio Pontos**
1	Ferramenta A	100	150
2	Ferramenta B	200	400
3	Ferramenta C	300	600
4	Ferramenta D	400	1200
5	Ferramenta Enterprise Architect da Sparks	500	3000
6	Ferramenta Requisite Pro da IBM	600	6000

Lista de Ações do evento: AoComprarCoisa()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Coisa Comprada **ID Coisa comprada**: (Ordem do Item)

9	**Imagem de Fundo**: img_praia **Obrigatória**: Sim. **Bloqueada**: Sim. **Pode voltar ao cenário principal**: Sim. **Permite Reiniciar**: Sim **Número de Tentativas Máximo**: 6 **Info Fase**: Área Praia **Tipo Fase**: Selecionar as Opções Corretas **Forma**: Lista de Múltipla Escolha **Lista Aleatória**: Sim **Descrição do Problema**: "O que Jack deverá fazer para sair da ilha com todos os Nerds e os Geeks" **Créditos Fase Finalizada**: ""

Ordem	**Opção**	**Correta?**
1	Verificar quais os requisitos que foram afetados pela mudança	Sim
2	Manter os requisitos como eles foram definidos por UbaapNerd e informar aos Nerds para aumentarem o tamanho do barco	Não
3	Incluir requisitos para acomodar as mudanças solicitadas	Sim
4	Informar a UbaapNerd sobre o impacto de tempo e de custo para realizar a mudança solicitada	Sim
5	Solicitar a aprovação de UbaapNerd depois dos requisitos já terem sido alterados	Não

Lista de Ações do evento: AoIniciarNaoExecutada()	
Seq. Ação	Definição
1	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dica(12) **Ativo:** Sim
2	**Tipo:** Disponibilizar objeto do baú **Objeto:** Bateria do Notebook **Ativo:** Se não existir
3	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dias restantes **Ativo:** Sim
4	**Tipo:** Disponibilizar objeto do baú **Objeto:** Dado da Sorte **Ativo:** Se existir

Lista de Ações do evento: AoExecutar()	
Seq. Ação	Definição

1	**Tipo**: Inicializar fase. - Exibir imagem de fundo

Lista de Ações do evento: AoReiniciar()	
Seq. Ação	Definição
1	**Tipo:** Remover Item do Baú **Objeto:** Dias restantes ➔ Como é inteiro, vai apenas diminuir um. Caso não possa mais fazer isso então executa ação associada ao item na configuração do baú.

Lista de Ações do evento: AoFinalizarSucesso()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa:** Fase executada
2	**Tipo:** Adicionar Recompensa ao Baú **Tipo Recompensa**: Pontuação Beneficio Coisa Comprada **Quando**: Possuir coisa comprada fase **Fase**: 8
3	**Tipo:** Exibir Créditos Fase Finalizada **Evento**: aoClicar() – fechar_popup
4	**Tipo:** Finalizar Cenário

10

Imagem de Fundo: img_aviao
Obrigatória: Não.
Bloqueada: Não.
Pode voltar ao cenário principal: Sim.
Info Fase: Área Avião
Permite Reiniciar: Não
Número de Tentativas Máximo: ND
Tipo Fase: Apenas Visitar

Lista de Ações do evento: AoIniciarNaoLiberada()	
Seq. Ação	Definição
1	**Tipo:** Adicionar Recompensa ao Baú **Tipo de Recompensa:** Recompensa("Bateria do Notebook)
2	**Tipo:** Exibir Popup **Descrição:** "O avião de Jack está muito danificado! Jack precisará encontrar outra forma de sair da ilha. Mas uma boa notícia: você encontrou nos destroços uma bateria carregada para o notebook de Jack" **Evento:** aoClicar() – fechar_poppup

APÊNDICE C – Realização de Caso de Uso e Diagrama de Classes de Negócio.

Um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos existentes entre eles. Os diagramas de classes também mostram as propriedades e operações de uma classe, além das restrições referente aos relacionamentos entre os objetos. No presente trabalho os diagramas de classes que estão disponíveis são a nível de domínio de aplicação e não pretendem resolver questões de arquitetura como a utilização de padrões de projeto ou técnicas de MVC.

O Projeto foi elaborado o Enterprise Architect (SPARKX, 2011) e está disponível em *https://drive.google.com/file/d/0B3SrSM9jH90jUUl1NWlNUTV3VFU/edit?usp=sharing*.

Para uma melhor visualização das classes, o projeto fora dividido em 4 pacotes: (1) Administração Básica; (2) Elaboração de Jogos; (3) Configuração do Cenário; e (4) Execução de Jogos. O diagrama de classes visualizado na Figura 25, representa o conjunto de classes do pacote de Administração Básica.
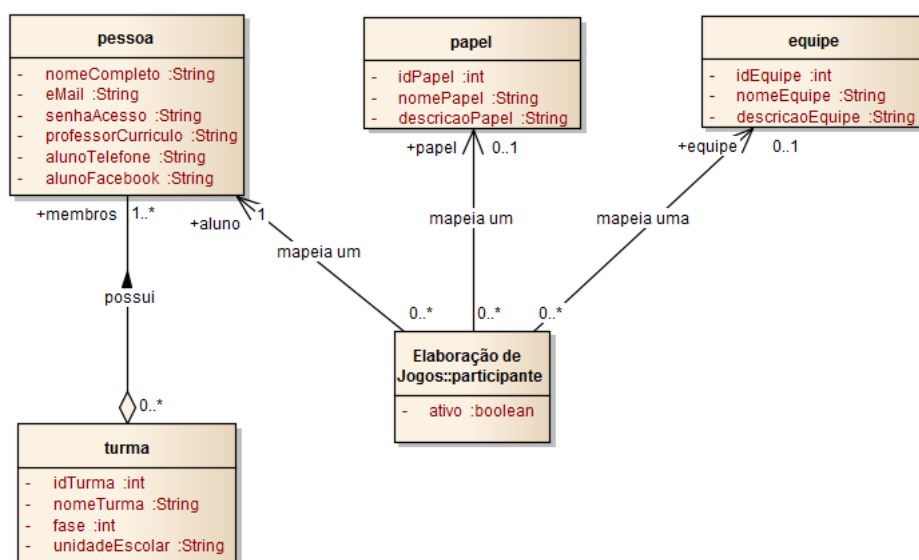
Figura 25 – Diagrama de Classes do Pacote de Administração Básica (Acervo do Autor).

Já na Figura 26 é exibido o diagrama de classes do pacote de execução de Jogos.
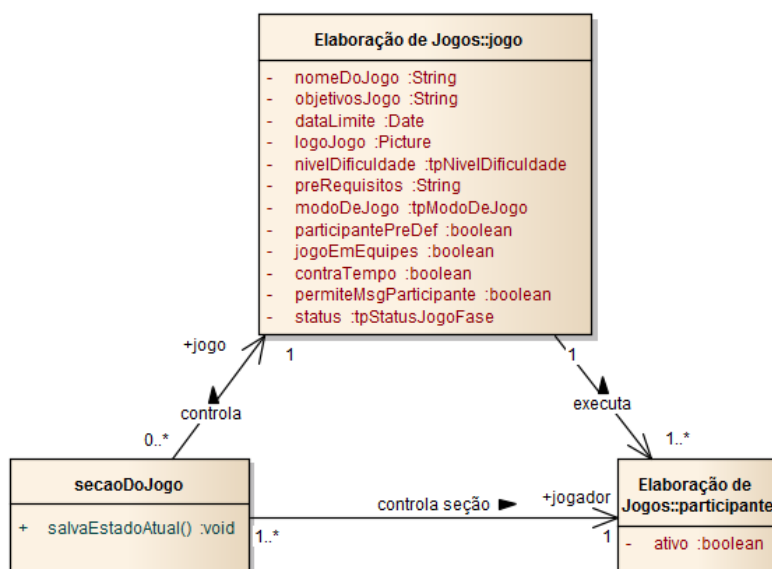
Figura 26 – Diagrama de Classes do Pacote de Execução de Jogos (Acervo do Autor).

A Figura 27, representa o diagrama de classes do pacote de elaboração de jogos. Ele apresenta a classe principal do gerador que é a *jogo*. O esquema utilizado em ConfigItemRecompensaDica, é uma forma de permitir a criação de tipos de classes para configuração dos mais variados itens de recompensa, conforme a demanda do gerador vai aumentando.

Uma atenção especial foi quanto aos tipos enumerados, para que fique bem claro quais são as constantes e do que cada um se refere, bem como onde é utilizada.
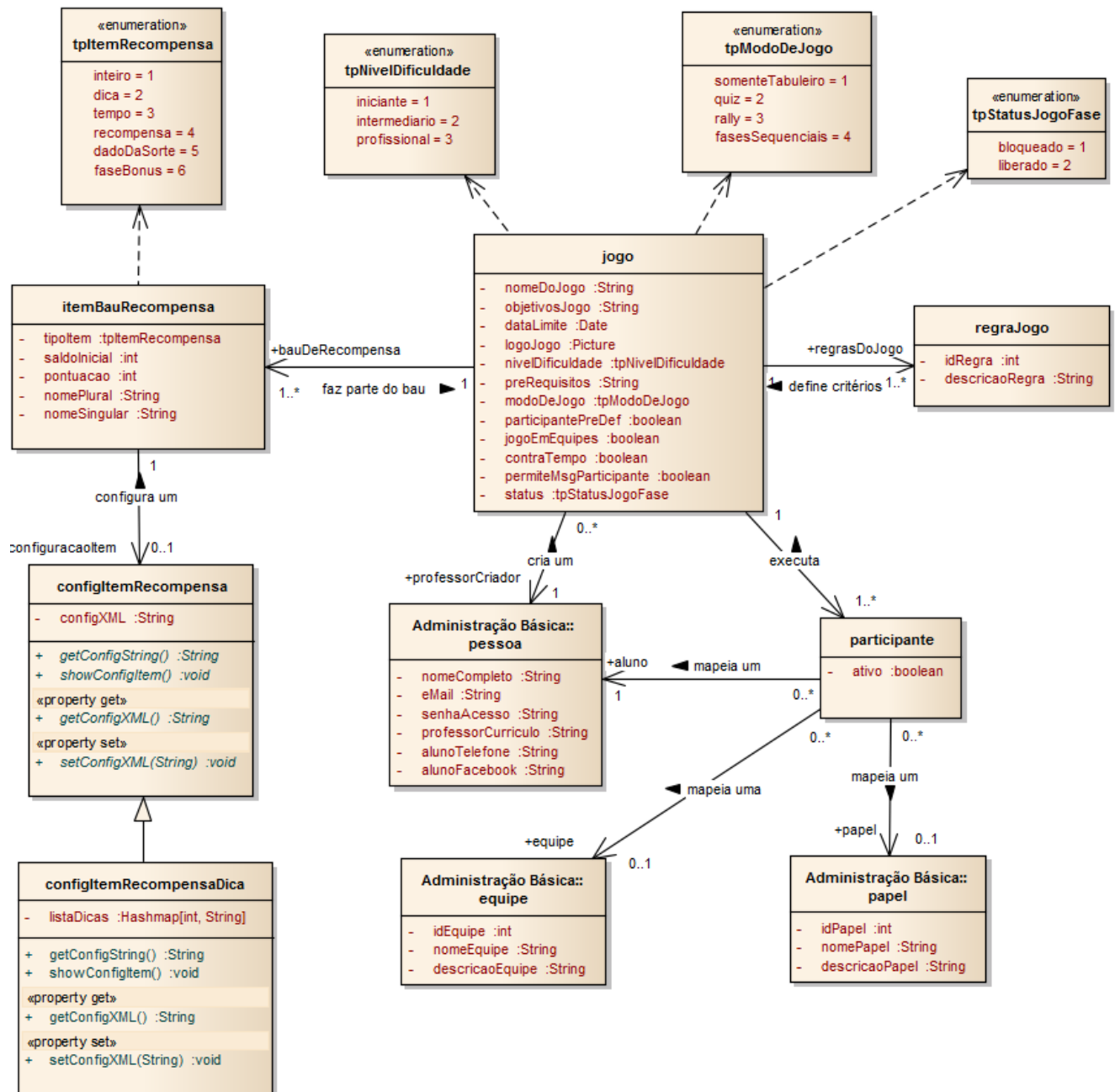
Figura 27 – Diagrama de Classes do Pacote de Elaboração de Jogos (Acervo do Autor).

Na Figura 28 está sendo exibido o diagrama de classes do pacote de configuração do cenário.
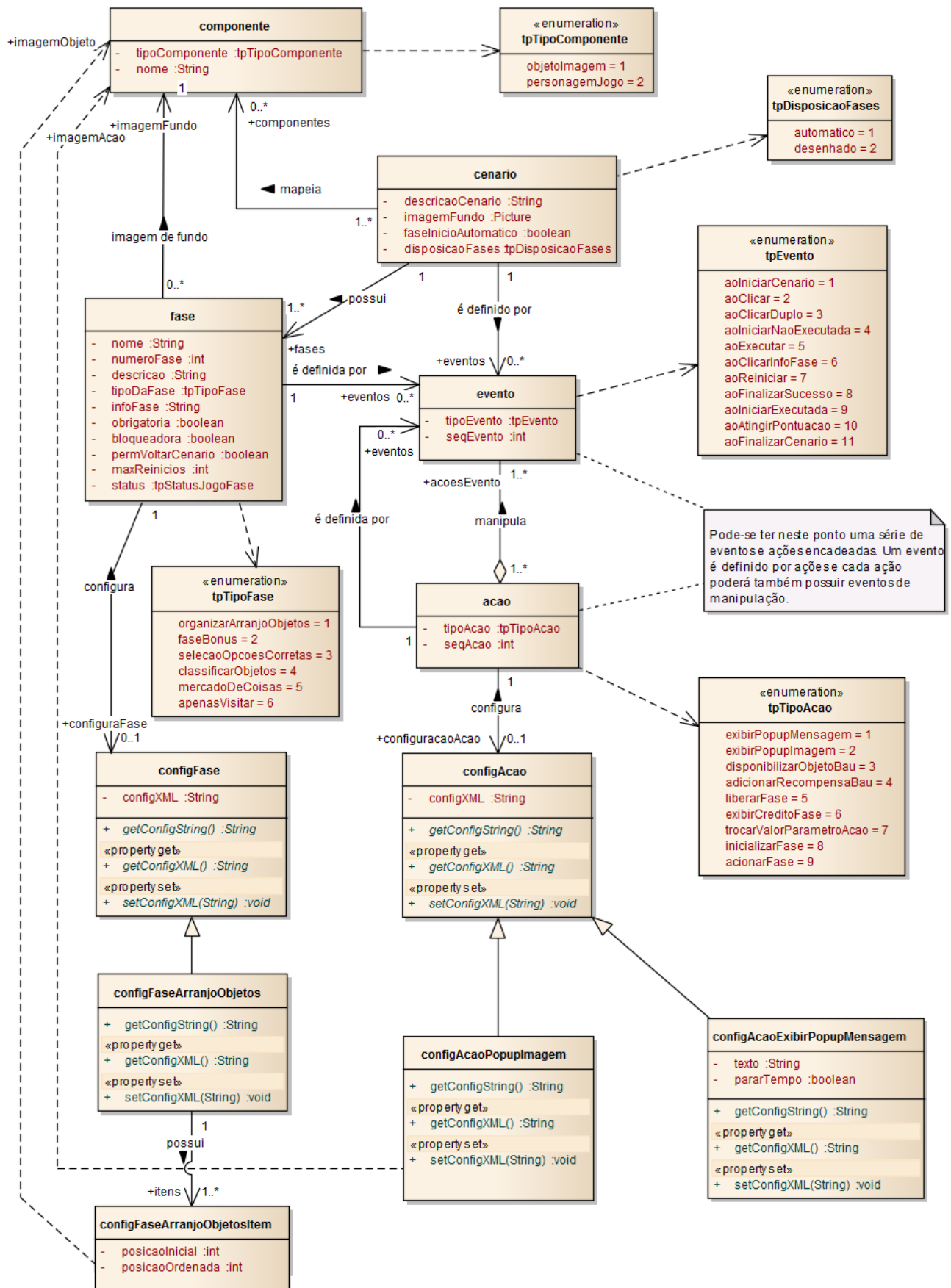
Figura 28 – Diagrama de Classes do Pacote de Configuração do Cenário (Acervo do Autor).

Os diagramas de interação descrevem como grupos de objetos colaboram com algum comportamento. Para a realização dos casos de uso foram utilizados os diagramas de interação de Comunicação e de Sequencia.

Para efeito de demonstração fora selecionado o caso de uso UC7 – Configuração do Cenário.

Utilizando-se o Enterprise Architect (SPARKX, 2011), na medida em que os casos de uso são realizados é inevitável a definição de classes a nível de projeto. Neste caso optou-se pelo modelo MVC (Model View Controller) além da persistência. Para tanto os pacotes de classes de projeto: controle, formulário, entidade e persistência foram definidos.

O diagrama de comunicação enfatiza os vínculos de dados entre os vários participantes da interação. Ele permite livre posicionamento dos participantes e permite desenhar vínculos para exibir como eles se conectam.

A Figura 29, mostra o diagrama de comunicação da realização do caso de uso UC7 – Configuração do cenário.
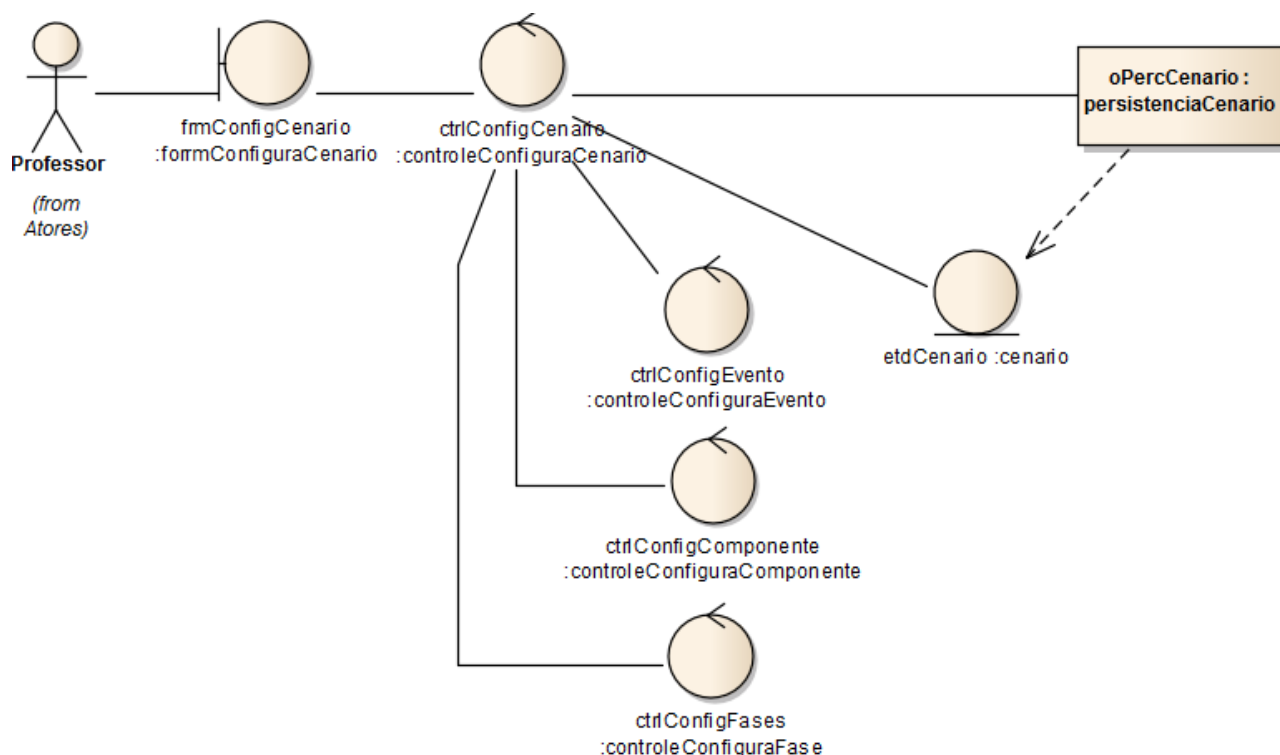
Figura 29 – Diagrama de Comunicação do UC7 – Configuração do Cenário (Acervo do Autor).

A seguir são apresentados os diagramas de sequencia, normalmente um diagrama de sequencia captura o comportamento de um único cenário. O diagrama mostra vários exemplos de objetos e mensagens que são passadas entre esses objetos dentro de um caso de uso.

Para a elaboração dos diagramas de sequencia da realização dos casos de uso, fora necessária a confecção dos cenários dos casos de uso.

Na Figura 30 é exibido o diagrama de sequencia da realização do caso de uso UC7 – Configuração do cenário.
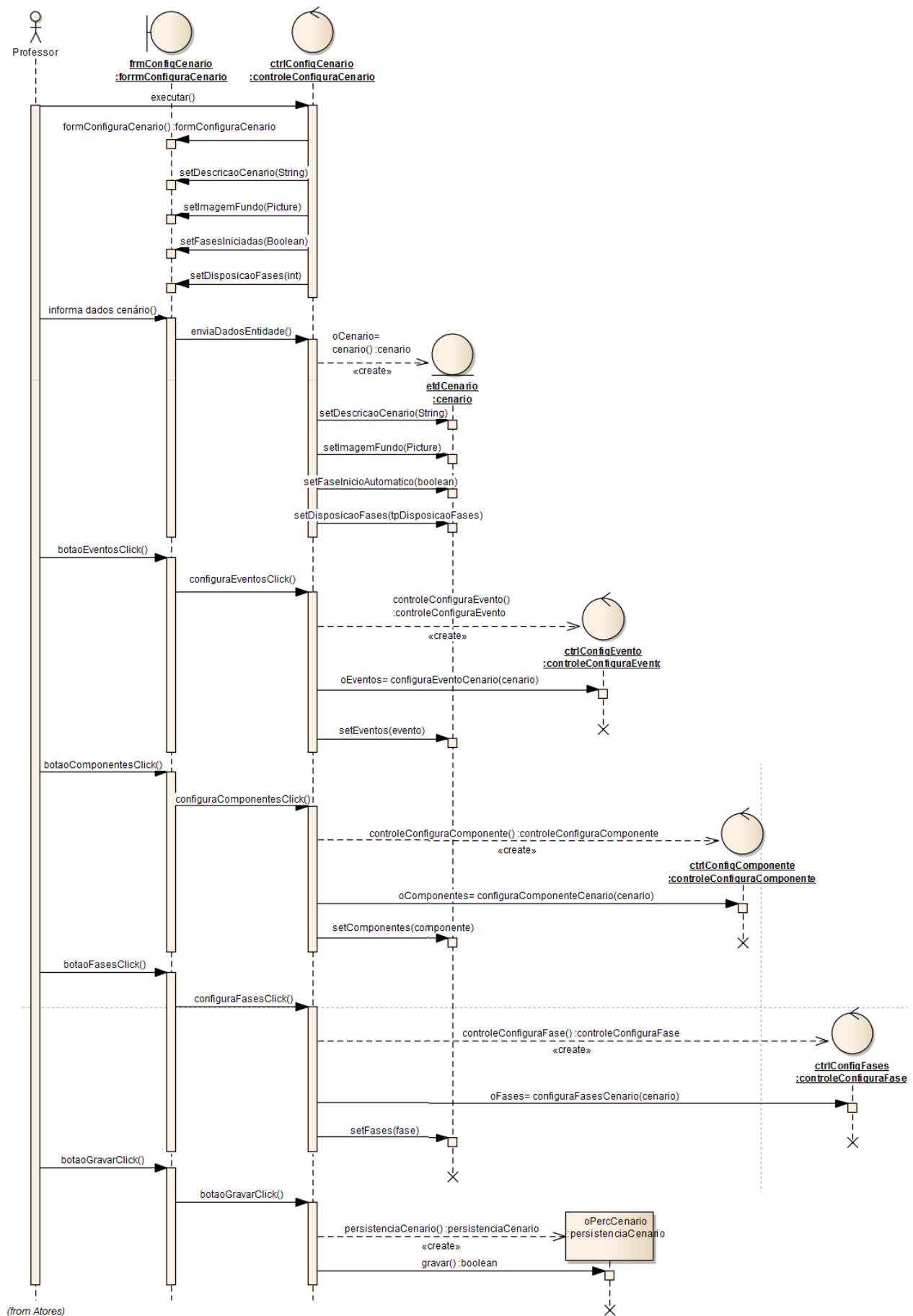
Figura 30 – Diagrama de Sequencia do UC7 – Configuração do Cenário (Acervo do Autor).